Secure Name Services for the Internet of Things

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum politicarum
(Doktor der Wirtschaftswissenschaft)

eingereicht an der
Wirtschaftswissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von
Herrn Diplom-Mathematiker Benjamin Fabian
geboren am 29.1.1971 in Berlin

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Dr. h.c. Christoph Markschies

Dekan der Wirtschaftswissenschaftlichen Fakultät:
Prof. Oliver Günther, Ph.D.

Gutachter:

1. Prof. Oliver Günther, Ph.D.
2. Priv.-Doz. Dr.-Ing. habil. Thomas Santen

eingereicht am:	26.06.2008
Tag der mündlichen Prüfung:	07.08.2008

Abstract

The term *Internet of Things* (IOT) describes an emerging global, Internet-based information service architecture for RFID-tagged items (Radio-Frequency Identification). In the vision of its proponents, this IOT will facilitate information exchange about goods in global supply chain networks, increase transparency, and enhance their efficiency. In an extension of this initial application scope, the IOT could also serve as backbone for *Ubiquitous Computing*, enabling smart environments to easily recognize and identify objects, and retrieve information from the Internet to facilitate their adaptive functionality; for example, in smart homes where items or furniture of daily use could be upgraded to provide information and counseling services.

Name Services for the IOT are distributed systems that serve the following fundamental lookup function: Given an identifier for a real-world object, e.g., an Electronic Product Code (EPC), they return a list of Internet addresses of services, which offer additional information about this object. Without name services acting as a broker between items and their information sources, the IOT could not achieve the flexibility and global scalability necessary to live up to its vision.

This thesis discusses the information security challenges involved in the design and use of an *IOT Name Service* (IOTNS), evaluates possible countermeasures to reduce security risks, and discusses fundamental trade-offs between performance and security. Our main contributions are the following:

- First, the requirements for an IOTNS are collected and discussed, including multilateral security and the client perspective, which have been neglected in IOT standards and research literature so far.
- Second, we conduct a detailed security analysis of the most influential standard *Object Naming Service* (ONS). This extends our previous article that initiated this new research line in the field of RFID and IOT security.
- Third, enhancements to ONS are discussed, which could mitigate some of the ONS security shortcomings in an evolutionary way without completely abandoning the established standard. In particular, we describe an architecture and prototype for Multipolar ONS, which reduces international dependency on a single country controlling the ONS Root.
- Fourth, we present a new IOTNS architecture based on Distributed Hash Tables (DHT) and its implementation on the research platform PlanetLab. This architecture is shown to offer enhanced overall security compared to ONS while delivering equivalent or even better functionality, scalability, and performance.

Future work should focus on the quantification of IOT diffusion and its scalability and performance demands, but also on further security requirements elicitation of its stakeholders, and on methods for secure and scalable cryptographic-key distribution among them. Emerging designs for IOT Discovery Services should take the security requirements, security and multipolarity analyses, as well as an extending of the DHT-based architecture presented in this thesis into consideration.

Keywords:
Internet of Things, RFID, Name Service, ONS, Security

Zusammenfassung

Mit dem Begriff *Internet der Dinge* (Internet of Things, IOT) wird eine im Entstehen begriffene globale, Internet-basierte Architektur von Informationsdiensten bezeichnet, die Informationen über mit RFID-Chips versehene Gegenstände bereitstellt (Radio-Frequency Identification). Nach der Vision seiner Befürworter wird das IOT den Informationsaustausch über Güter in in globalen Logistiknetzen erleichtern, ihre Transparenz erhöhen und somit Effizienzsteigerungen erreichen. Als eine Erweiterung seines ursprünglichen Anwendungsgebiets könnte das IOT auch als Rückgrat des *Ubiquitous Computing* fungieren und sogenannte intelligente Umgebungen in die Lage versetzen, Objekte leicht zu erkennen und zu identifizieren sowie Informationen aus dem Internet abzurufen, um damit ihre adaptive Funktionalität zu unterstützen. Ein Beispiel dafür sind intelligente Wohnumgebungen, wo Alltagsgegenstände und Möbel um Informations- und Beratungsdienste erweitert werden könnten.

Namensdienste für das IOT sind verteilte Systeme, die die folgende wichtige Suchfunktion bereitstellen: Bei Eingabe eines Identifikators für einen Gegenstand, z.B. eines Elektronischen Produktcodes (EPC), wird eine Liste von Internetadressen für Dienste zurückgegeben, die weitere Informationen über den Gegenstand anbieten. Ohne derartige Namensdienste, die als Vermittler zwischen Gegenständen und zugehörigen Informationsquellen dienen, könnte das IOT nicht den Grad an Flexibilität und globaler Skalierbarkeit erreichen, der zur Erfüllung seiner Vision notwendig ist.

Die vorliegende Arbeit hat die Herausforderungen an die Informationssicherheit zum Thema, die mit Entwurf und Nutzung von IOT-Namensdiensten (IOTNS) verbunden sind, evaluiert mögliche Gegenmaßnahmen, um ihre Sicherheitsrisiken zu reduzieren, und diskutiert grundsätzliche Abwägungen zwischen Sicherheit und Systemleistung. Hierbei leisten wir die folgenden Forschungsbeiträge:

- Erstens werden die Anforderungen an einen IOTNS herausgearbeitet, wobei insbesondere mehrseitige Sicherheit und die Perspektive der IOTNS-Clients berücksichtigt werden, die in den Standards und der Forschungsliteratur zum IOT bisher vernachlässigt worden sind.
- Zweitens führen wir eine Sicherheitsanalyse des einflußreichen Standards *Object Naming Service* (ONS) durch. Diese Analyse erweitert unsern früheren Artikel, der diese neue Forschungslinie im Bereich der RFID- und IOT-Sicherheit begründete.

- Drittens werden Verbesserungen des ONS diskutiert, die einen Teil der ONS-Sicherheitsprobleme beheben könnten, ohne den etablierten Standard vollständig zu verändern. Hierbei werden insbesondere eine Architektur für Multipolares ONS und ihr Prototyp vorgestellt, bei der die internationale Abhängigkeit von dem Land reduziert werden kann, das den ONS-Root kontrolliert.
- Viertens präsentieren wir eine neue IOTNS-Architektur und ihre Implementierung auf der Forschungsplattform PlanetLab, die auf verteilten Hashtabellen (Distributed Hash Tables, DHT) basiert und von der gezeigt wird, dass sie verbesserte Sicherheitseigenschaften gegenüber ONS aufweist – bei vergleichbarem oder sogar erhöhtem Grad an Funktionalität, Skalierbarkeit und Systemleistung.

Weiterführende Forschung sollte ihren Fokus auf die Verbreitung des IOT und eine Quantifizierung seiner Skalierbarkeits- und Leistungsanforderungen richten, aber ebenso auf eine weitergehende Analyse der Sicherheitsanforderungen der beteiligten Akteure sowie auf Möglichkeiten, kryptographische Schlüssel sicher und skalierbar unter ihnen zu verteilen. Entwürfe für zukünftige *Discovery Services* sollten die in dieser Arbeit herausgearbeiteten Sicherheitsanforderungen und Analysen zu Sicherheit und Multipolarität berücksichtigen sowie eine Weiterentwicklung der vorgestellten DHT-basierten Architektur in Betracht ziehen.

Schlagwörter:
Internet der Dinge, RFID, Namensdienst, ONS, Sicherheit

ἡ μὲν θαμβήσασα πάλιν οἶκόνδε βεβήκει·
παιδὸς γὰρ μῦθον πεπνυμένον ἔνθετο θυμῷ.

Dedicated to My Mother

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

The term *Internet of Things* (IOT) describes a collective, global, Internet-based information service architecture for items equipped with RFID tags (Radio-Frequency Identification). In the vision of its proponents, the IOT will increase transparency and facilitate information exchange about goods in global supply chain networks, and enhance their efficiency. In a broadening of its initial application scope, however, the IOT could also serve as the backbone for *Ubiquitous Computing*, enabling smart environments to ascertain objects and recognize people, and retrieve information from the Internet to facilitate the adaptive functionality they provide; for example, in smart homes where kitchen appliances or furniture for everyday use can be enhanced to provide information and counseling services.

Name Services for the IOT serve the following lookup function: Given an identifier for a real-world object, e.g., an Electronic Product Code (EPC), return a list of Internet addresses of services, which offer additional information about this object. This thesis discusses the information security challenges involved in the design and use of an *IOT Name Service* (IOTNS), and evaluates possible countermeasures to reduce their security risks.

1.2 RFID and the Internet of Things

Radio-Frequency Identification (RFID) is a communication and identification technique known at least since the Second World War from *friend-or-foe* identification systems of military airplanes.[1]

In recent years, however, RFID is used in many new civil application fields – ranging from animal or human identification, anti-counterfeiting, access control and

[1] Rieback et al., 2006 [170].

payment, to global supply chains,[2] finally reaching the area of smart environments, *Pervasive* or *Ubiquitous Computing*, and so-called *Ambient Intelligence*.[3]

The term *Internet of Things*, as it is established within RFID and supply chain communities today, describes the collective global information service architecture for RFID-tagged items; that is, networked services that *speak about* things, rather than services that reside *inside of* the objects themselves.[4] This meaning of *Internet of Things* – with emphasis on globally distributed RFID-information service architectures – will be adopted in this thesis, and will be abbreviated by IOT.

In the near future, mainly due to monetary, energy, and space costs, it is expected that most "things" will only be equipped by simple chips called *Tags*, which are mostly externally powered and are communicating via radio waves issued by *RFID Readers*.[5] An RFID reader interrogates all tags in its vicinity via radio waves of a specific frequency, in the case of passive chips also providing energy for an answer in the same process. The reader and tags follow an anti-collision protocol (*Tag Singulation*) to establish an answering order. Then, the tags return the data requested by the reader.[6]

These tags are in general[7] only capable of storing little data (e.g., identification numbers), and can only process simple operations. The "intelligence," the decision making, business processes, adaptivity, and not least the information storage and retrieval will all happen at the back-end, at a middleware or application layer, and also via the Internet, e.g. by the use of Web services offered by many different parties – a paradigm known as *Data on Network* (versus *Data on Tag*).[8]

1.3 Electronic Product Code

Besides the anticipated ubiquity of RFID tags and readers, there is another important factor that facilitates the establishment of an *Internet of Things*: The standardization of a global numbering scheme for physical objects, the Electronic Product

[2] Garfinkel and Rosenberg, 2005, pp. 381 [71]; Bullinger and ten Hompel, 2007 [24].

[3] Fabian and Hansen, 2006 [60].

[4] The Internet of Things in this RFID-specific sense is not an *Internet* from the classical computer network perspective, which would comprise an inter-network of smaller local networks, cf. Tanenbaum, 2003, p. 25 [199]. See also Liu and Albitz, 2006, p. 2 [122]. Each of those would consist of nodes capable of autonomously participating in the network, for example through the use of a fully-grown Internet Protocol (IP) stack such as IPv6 with its vast address space and support for mobility, see Loshin, 2004, pp. 291 [126].

[5] Finkenzeller, 2006, Ch. 3 [65]; for the important standard UHF Class-1 Gen 2, cf. EPCglobal, 2007 [52].

[6] For a detailed description of the inner workings of RFID systems, the standard technical reference is Finkenzeller, 2006 [65].

[7] There are definitions for classes of more powerful RFID tags, but those are not as useful yet for use on most retail items due to their size, cost, and energy consumption.

[8] For a comparison and evaluation of these paradigms, see Diekmann et al., 2007 [42].

Code (EPC). If the vision of many RFID proponents becomes reality, more and more common objects will soon acquire some kind of cyber presence. Objects will be equipped with RFID tags containing identification data and possibly some additional information about the object in question.

To keep tag costs low, one may often merely store an identifier and use it as a key to access databases containing the actual object information. This second approach is typical for the important *EPC Tags* – RFID tags that aim to replace the conventional barcode system. This EPC, which is globally unique, can be used as a key to retrieve information from the *EPCglobal Network*, a widely distributed system of databases.[9] The EPC standard represents a numbering framework that is independent of specific hardware features, such as tag generations or specific radio frequencies. This influential numbering system is about to enhance and finally replace traditional bar codes. It aims to assign a globally unique number to nearly every object equipped with an RFID tag. This EPC is serving as an identifier for the physical object carrying the tag, which can now be recognized, identified, and tracked by an IT infrastructure.

1.4 EPC Tag and Data Standards

EPC tags are potentially the most important class of RFID tags, and constitute the physical embodiment of the EPC to be attached or integrated into supply chain pallets and transporting cases, and possibly to all applicable manufactured single goods of the future.[10] Though the EPC standard is actually a meta framework for different encoding schemes and name spaces, most EPCs have a structure similar to the one shown in Fig. 1.1, which depicts an example EPC for one of the most popular standards, the *Serialized Global Trade Identification Number* (SGTIN).[11]

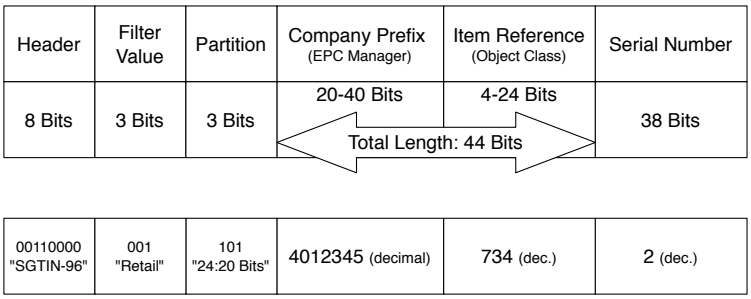


Figure 1.1: SGTIN-96 EPC

In this SGTIN-96 variant, the EPC includes a *Header* to denote its EPC identity type (here: SGTIN-96), a *Filter Value* for fast logistic decisions, a *Partition Value*

[9] EPCglobal, 2007 [53].
[10] The official document for EPC data standards is currently EPCglobal, 2007 [51].
[11] EPCglobal, 2007, pp. 26, 88 [51].

that indicates the boundary of the next two fields, and a *Company Prefix* (also referred to as *EPC Manager*) that is a unique identifier of the item manufacturer. Furthermore, the manufacturer can assign *Item Reference Numbers* (also called *Object Classes*, OC) to classes of objects she produces. Within the same class, similar objects can be distinguished by their *Serial Number* – this is a fundamental extension compared to the conventional barcode. Other EPC numbering systems besides GTIN-96 are shown in Figure 5.10 in Chapter 5.[12]

In the following, important application fields of the *Internet of Things* are discussed.

1.5 Supply Chains

Of the many potential application areas for RFID, EPC, and the *Internet of Things*, we present two important fields: supply chains and smart homes. The first, because the supply chain can be considered as the main driver of RFID and EPC adoption. The second, because if the IOT is in place for the supply chain, it could quite naturally extend to end-user or consumer services using the item information infrastructure already in place.[13]

Cost pressure and transparency demands are main drivers for the adoption of RFID in the supply chain.[14] At the current time, many RFID pilot projects focus on intra-organizational use, that is, optimizing manufacturing processes within one company. However, there are indicators of strategic advantages of item information flow between companies, which could optimize the whole supply network, not only its nodes. This information sharing between companies could be enabled by the IOT and the EPCglobal Network.[15]

In Fig. 1.2,[16] the path of an RFID-equipped item through a supply chain is depicted. At every station – manufacturer, suppliers, shop – the EPC is read by RFID readers and stored in local databases together with context information – time, location, physical environment conditions, or business process steps. By subsequently retrieving this data, the item's path through the chain becomes transparent, inventorying becomes easier, bottlenecks could be identified, and handling processes be optimized. Currently, RFID-tagging is mostly used at the container and pallet level; however, in part due to massive investments of influential companies, future tagging of most consumer items is to be expected.

[12] Image source: EPCglobal, 2007, p. 90 [51].

[13] This probable instance of *Innovation Diffusion* (Rogers, 2003 [173]) needs more future study, because it is subject to several possible constraints, for example openness and cost of participation in the IOT, as well as scalability and performance.

[14] Fleisch and Mattern, 2005 [68].

[15] Leong et al., 2004 [119]; VeriSign, 2005 [203]; Wamba et al., 2006 [207]; Wamba and Boeck, 2008 [206].

[16] Adapted from EPCglobal, 2004, p. 7 [49].

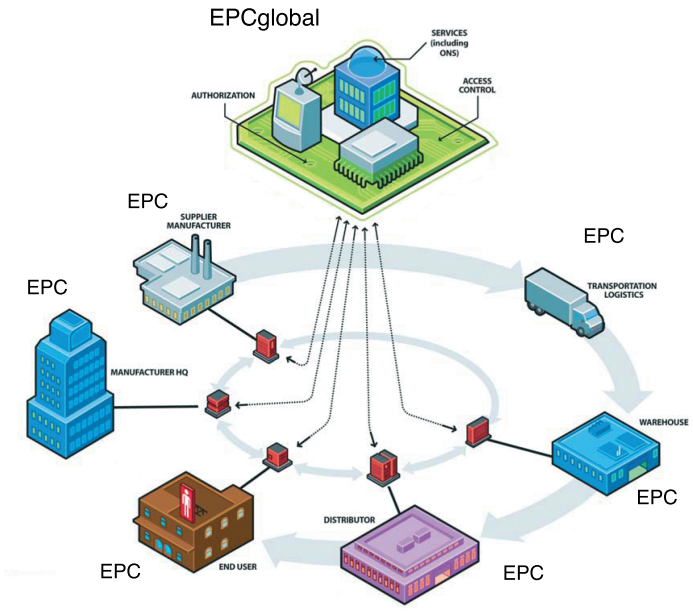


Figure 1.2: EPC in the Supply Chain (Source: EPCglobal)

On the shop-floor, automatic and nearly real-time inventorying would be possible, as well as customer profiling, tracking, and new recommender systems. Item-level tagging could facilitate reverse supply chains for returned goods, and could also enable after-sale services, coupled with smart home applications.

1.6 Smart Homes

RFID is also a key enabling technology for so-called *smart environments*, physical surroundings – such as cars and houses – enhanced by a multitude of networked devices, which are currently developed by many researchers and companies, gradually realizing early visions on *Ubiquitous Computing* (UC) or *Ambient Intelligence*.[17] Even if other sensor technology and image recognition advances, RFID will offer simple, effective, and cheap operations suitable for the mass market.

The general model of an UC system is depicted in Fig. 1.3.[18] RFID readers will function as a sensor and identification layer that feeds data into an adaptive decision making engine. This engine consults internal, external, or even Internet data sources

[17] Weiser, 1991 [213]; Mattern, 2003 [128]. For RFID and UC, cf. Floerkemeier et al., 2004 [69]; Liu et al., 2006 [123]. For a more in-depth presentation of UC technologies, cf. Fabian and Hansen, 2006 [60].

[18] This figure is cited from a TAUCIS chapter [60], and was derived in cooperation with the FIDIS project (`http://www.fidis.net`) (03.2008).

to decide what and how a service can be delivered back to the entity to which the system adapts.

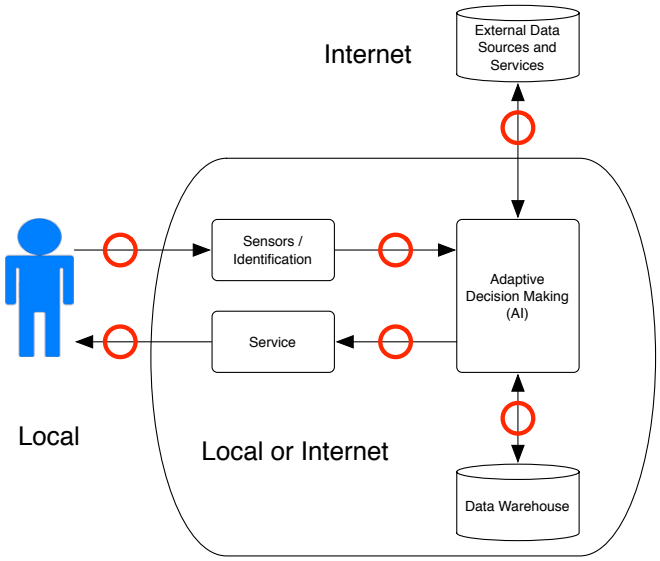


Figure 1.3: General Model of an UC System

Examples of home applications are so-called smart shelves and refrigerators that know their own inventory, and could enable services for search, delivery, food or health counseling.[19] Testbeds for future supermarkets, smart factories, and home applications are for example the Gator Tech Smart House[20] and the METRO RFID Innovation Center.[21] Smart office buildings are already becoming commonplace, today.[22]

Supply chain and smart home applications share many requirements. Both, as well as other EPC-aware applications, raise the following question: How can distributed data sources be located on the Internet, which correspond to a given EPC? This is the task of a *Name Service*.

1.7 Name Services

Name services, in their fundamental function, translate strings, such as human-memorizable names, into network identifiers that can be used for message routing, for example – and most prevalent today – into IP addresses. In addition, name services may offer further information that is related to the name being queried for, for example corresponding mail servers or public-key records.

[19] Stajano, 2002. p.51 [192]; Fabian and Hansen, 2006 [61]; Rothensee, 2008 [174].
[20] Helal et al., 2005 [88]. URL: `http://www.icta.ufl.edu/gt.htm` (03.2008).
[21] URL: `http://www.future-store.org` (03.2008).
[22] Ivanov et al., 2007 [95].

Classic examples for name services are distribution services for `/etc/hosts` files, NIS, NetBios, WINS, and most importantly, the Domain Name System (DNS).[23]

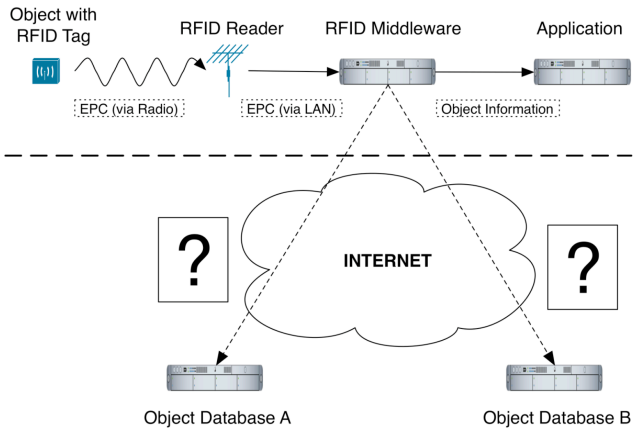


Figure 1.4: Function of an IOT Name Service

With respect to the *Internet of Things*, a name service helps in the following situation (Fig. 1.4): We have read an identifier, regarded as a *name*, from an RFID tag attached to an object – how do we find corresponding information sources on the Internet? In short, an IOT name service resolves object identifiers to information service addresses.

In the main reference architecture by EPCglobal[24] two categories of name services are used. The first is the *Object Naming Service* (ONS) to locate the item manufacturer and uses – as of today – only EPC Manager and Object Class fields of an SGTIN EPC. The second category comprises *EPCIS Discovery Services*, which shall offer lookup of multiple information sources related to fully-serialized EPCs.

1.8 Security

Given the main application areas of the IOT – to serve as a critical business infrastructure and as a possible information support for Ubiquitous Computing permeating society – there will be tremendous demand for making the process of using the IOT safe and secure.

Given their pivotal role as information brokers, IOT name services must perform reliably and securely in the face of errors and random disruptions common on the Internet, but also in the face of malicious entities threatening to disrupt the IOT functionality, modify information for staging further attacks, or use the data that is transmitted – or generated in IOTNS transmission logs – for their own purposes.

[23] More details on DNS will be presented in Section 3.3.2.
[24] See Chapter 3; also cf. EPCglobal, 2007 [53].

This thesis investigates the following topics: What are the security requirements that an IOT name service must fulfill? What security problems exist with the currently proposed name service ONS, and could ONS be made more secure with respect to particular requirements and threats?

Finally, we will present an affirmative answer to the question whether there are alternative architectures to ONS that would satisfy many security requirements better.

1.9 Thesis Contributions and Outline

This thesis provides the following contributions:

- First, the requirements for an IOTNS are discussed in a systematic fashion, including client requirements on multilateral security, which have been neglected in the IOT standards so far.

  This is based on the following publications: Fabian, Spiekermann, and Günther, 2005 [62]; Bauer, Fabian, Fischmann, and Gürses, 2006 [12]; Fabian, Gürses, Kuzmanovski, and Santen, 2006 [63]; Fabian and Hansen, 2006 [59]; Fabian and Günther, 2007 [57].

- A detailed security analysis of the most influential IOTNS standard, the *Object Naming Service* (ONS), is conducted.

  This is based on the first published security analysis of the ONS, Fabian, Spiekermann, and Günther 2005 [62], and on the discussion of the confidentiality challenges of EPCglobal Network as a whole, to be published in Fabian and Günther, 2009 [58]. The first article initiated this new research line in the field of RFID and IOT security and stimulated public discussions on ONS in the EU.

- The first formulation and discussion of the Multipolarity requirement for ONS is presented, in conjunction with Multipolar ONS (MONS), a corresponding modification to ONS that guarantees multipolarity (joint work).

  Publication: Evdokimov, Fabian, and Günther, 2008 [55].

- An analysis of possible security extensions and their applicability to ONS and EPCIS is presented.

  Publication: Fabian and Günther, 2009 [58].

- The presentation of a P2P-based alternative to ONS (OIDA), which takes multilateral security requirements into account.

  Publication: Fabian and Günther, 2007 [57].

- The implementation and testing of OIDA on the international research network PlanetLab, presenting empirical evidence for the feasibility of P2P-ONS with respect to IOTNS functional and performance requirements (publication in preparation).
- A security analysis of P2P-ONS in general – and OIDA in particular – is presented, and additional security measures and their adaptation to OIDA are discussed (publication in preparation).

This thesis is structured as follows. In the current Chapter 1, we have presented an introduction to the IOT, the problem statement, and the main contributions. Chapter 2 will collect the requirements that an IOT name service should fulfill, with a special emphasis on confidentiality requirements of the clients. Using those requirements as a foundation, Chapter 3 will discuss the ONS proposal by EPCglobal, and a corresponding security analysis will be conducted. In Chapter 4, we will investigate if and how ONS could be made more secure with respect to particular requirements, without changing the initial design too much. A special section will be dedicated to the feasibility of Multipolar ONS.

With Chapter 5, a paradigm shift from Client-Server to Peer-to-Peer systems for IOTNS will be conducted. An alternative IOTNS architecture called OIDA based on Distributed Hash Tables (DHT) will be presented. We will discuss a prototypical implementation and experimental results on OIDA's feasibility and performance. In addition, we will compare the security properties of OIDA in several scenarios of IOT adoption and key distribution to the requirements identified in Chapter 2. A comparison of the IOTNS architectures will close this chapter.

Finally, Chapter 6 will summarize the results of this thesis, closing with an outlook on open research problems.

Chapter 2

Name Service Requirements

> The Indians of Chiloe keep their names secret and do not like to have them uttered aloud; for they say that there are fairies or imps on the mainland or neighbouring islands who, if they knew folk's names, would do them an injury; but so long as they do not know the names, these mischievous sprites are powerless.
>
> *Sir James Frazer*
> The Golden Bough[a]

[a] Sir James Frazer: The Golden Bough. Wordsworth, 1993 (1922), p. 245.

2.1 Introduction

Before a system can be built, it should be clear what it aims to achieve, indicated by a collection of its requirements. The requirements for an IOTNS will be discussed in a systematic fashion, including client requirements on multilateral security, which have been mostly neglected in the IOT standards and related literature so far. The requirements gathered in this chapter will serve as a guiding framework to compare different IOTNS architectures in later chapters of this thesis.

First we will discuss related work for this chapter in the following. Some functional and scalability requirements for an IOTNS have been gathered by EPCglobal, and are discussed in the ONS specification.[1] Relevant work on requirements elicitation for EPCIS Discovery Services has been conducted by the EU BRIDGE project together with GS1, including interviews of a small number of companies.[2] The requirements collected focus on functional and performance aspects, availability, integrity, as well as provider data confidentiality, but are rather neglecting the client's perspective. Another recent line of research on Discovery Services is presented in

[1] Mealling, 2005 [129].
[2] BRIDGE, 2007, pp. 8 [22]. BRIDGE, 2007 [23].

Kürschner et al., 2008 [115], where many similar requirements to those presented in the current chapter are identified, including a joint requirement on provider and client confidentiality.[3] While presenting a peer-to-peer alternative to the DNS, Ramasubramanian and Sirer, 2004 [161], have collected a short set of functional, performance, and robustness requirements for general name services, which have been adapted and extended in this chapter.

This chapter is structured as follows. First, functional and performance requirements for an IOTNS are presented. Then security requirements are discussed, with an emphasis on motivating the client's need for confidentiality while using an IOTNS – and the IOT in general.

2.2 Functional and Performance Requirements

What are the requirements a name service for the IOT should fulfill? The following sections do not aim to reflect the whole current research on requirements engineering. Even the term *requirement*, whose interpretation ranges from high level goals – adopted here – to detailed formal system specifications in the literature, cannot be discussed in depth here. Instead we focus on an informal discussion of the most basic needs stakeholders of an IOT name service (IOTNS) would like to see fulfilled.

This allows us to establish a set of design guidelines, as well as evaluation criteria to compare different IOTNS architectures. Requirements and security engineering both constitute iterative processes.[4] This process can only be covered partially for the IOT at this point in time where the success, diffusion, and application areas of the IOT, as well as the set and goals of its stakeholders, are not yet clearly discernible.

There is an established dichotomy of functional vs. nonfunctional requirements.[5] Functional requirements describe the functionality and services that a system should provide. Non-functional requirements are often considered constraints on the system functionality, such as performance, quality, safety, and security.

Here we present an essential set of *high-level* functional and non-functional requirements for an IOT name service S in natural language, extracted from literature and the analysis of other, existing name services. S is understood with Jackson as a *system*, defined as a the *machine* to be built, together with its *environment*.[6] Requirements can be fulfilled by the machine, the environment – such as assumptions on the application area and organizational procedures – or by their conjunction. Regarding

[3] Kürschner et al., 2008, p. 23 [115], but, unlike our work, without a further distinction of several kinds of confidentiality requirements.

[4] For iterations in software engineering cf. Sommerville, 2004, Ch. 4 [188]. For iterative processes in security engineering, cf. Anderson, 2001, pp. 498 [3].

[5] Sommerville, 2004, Ch. 6 [188].

[6] Jackson, 2001 [96].

the IOT environment, we for example require that there will be a *membership and authorization procedure* for clients and providers in place, which we consider not mainly a security requirement, but a fundamental functional requirement.

Some of the following requirements have been identified in Ramasubramanian and Sirer, 2004 [161], for alternatives to the DNS. Other sources and related work include the ONS specification [129] by EPCglobal, which however, rather indirectly presents the assumed requirements, and the Discovery Service requirements collected by the EU BRIDGE project [22].

To formulate the requirements in a general way, we will use the term *OID* (Object Identifier) in the following section, since an IOT name service should be able to serve not only EPC numbering schemes, but also arbitrary current or future object numbering systems.[7] The term Object Information Service (OIS) describes sources of actual object information and is a generalization of the EPCIS Information Services of the EPCglobal Network (see Chapter 3). We will use, however, the term EPCIS in later parts of the chapter on motivating security requirements where properties of the most influential IOT realization thus far, the EPCglobal Network, are reflected. Table 2.1 shows the high-level functional roles relevant for an IOTNS, as well as example stakeholders connected to these functions.

Functional Role	**Stakeholder**	**Example**
Object Information Service (OIS)	Information Provider (Publisher)	Manufacturer EPCIS
OIS Resolver	Client	Shop, Smart Home IT
IOT Central Node	IOT Infrastructure Provider	EPCglobal Core Service
IOTNS Node	Node Provider	ONS Server
IOTNS Special Node	Node Provider	ONS Root Server
OIS Discovery Service	Discovery Service Provider	EPCIS Discovery Service
Router	Internet Service Provider	Local or Backbone Router

Table 2.1: IOTNS Functional Roles

Figure 2.1 shows the IOTNS function in the context of the Internet of Things, distinguishing its basic function from the actual Object Information Services and the not yet specified Discovery Services that will possibly provide an overlap in functionality, but will in general also support more complex or long-standing queries.

The following high-level functional, scalability, performance, and robustness requirements for an IOT name service S can be identified.[8]

Functional Requirements for an IOT Name Service

1. System Membership and Authorization Procedure: A set of membership definition and authorization procedures for all publishers and clients of S shall be provided. This will be mainly be part of the environment, due to its organizational nature. Those procedures shall define:

[7] Note, that ONS in its specification of version 1.0 only works for SGTIN EPCs, see Mealling, 2005 [129].

[8] The following extends the previous presentation in Fabian and Günther, 2007 [57].

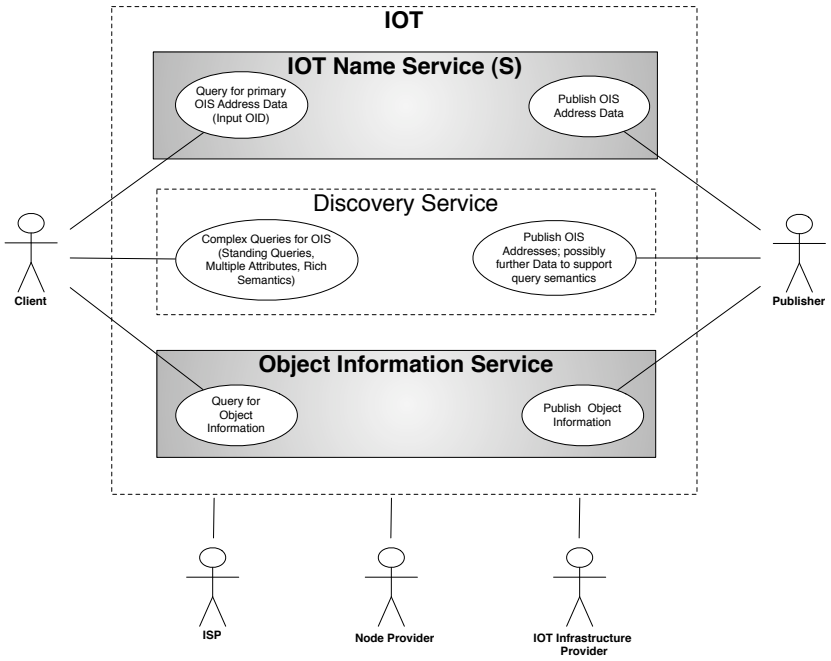


Figure 2.1: IOTNS Function in Context

(a) Which publishers shall be authorized to publish information about what kind of OIDs.

(b) Which clients shall be granted authorized access to what OIS address information and to which actual object information.[9]

(c) Which parties are allowed or even obliged to insert (machine) components into S (like servers or computer nodes) and who may run sub-services of S, for example security services in the form of Certification Authorities.

The first and third procedure should be a global convention between all IOT users, the second set of procedures can be delegated to the authorized information providers. Note that in theory, membership could be free to everyone, so that everyone may be able to publish and retrieve information about any object. In practice, there will be constraints by the information providers' economic and security interests.

2. Flexible OID Support: S should be flexible in its support for different OID schemes.

3. Publishing: An information provider shall be able to input address documents into S for OIDs for which he is authorized to publish information. These documents shall include addresses of OIS servers providing information about objects carrying those OIDs.

[9] Only if object information itself is stored in S to integrate OIS (cf. Section 5.7).

4. Multiple Publishers (independent): Multiple independent but authorized publishers should be able to provide information for an OID by storing corresponding address data in S, without possible mutual interference, like censorship.[10]

5. Querying: On input of OID e by a client, S shall output a current list of servers offering information about the object corresponding to e.

6. Updating: Authorized publishers shall be able to update the data records they published at will.

7. Deleting: Authorized publishers shall be able to delete the data records they published at will. A time-to-live value (TTL) should be provided for each document to indicate old data and to reduce overhead for deletion.

8. Class-level Addresses: If the OID is structured into a class-level and serial-level part, S shall be able to work with partial OIDs at the class-level; for example a partial SGTIN EPC consisting of EPC Manager and Object Class.

9. Serial-level Addresses: If the OID is structured into a class-level and serial-level part, S should be able to work with fully serialized OIDs, for example a complete SGTIN EPC consisting of EPC Manager, Object Class, and Serial Number (see Fig. 1.1).

10. Object Information (optional): S should itself be able to store and return (small amounts) of object information about OIDs to reduce query overhead, for example directly indicating if an object's official lifetime has expired.

Scalability

The system must be able to work on a global scale. Because it is used for the IOT, it is probable that S – in the long run – must cope with much more traffic than the usage of DNS for URL name resolution generates today.

1. High Node Count: S should work with a very large number of participating nodes (servers).

2. High Client Count: S should work with a very large number of participating clients.

3. Scalability to Medium IOT Adoption:[11] S shall work in scenarios with a medium level adoption of the IOT across businesses.

[10] This is currently not satisfied by ONS where EPC Managers control the ONS data, but is one of the goals for EPCIS Discovery Services.

[11] For an attempt to quantify the scale of IOT adoption scenarios, see Section 5.5.2.

4. Scalability to Large IOT Adoption – Class-level Lookups: S should work in scenarios with a high level adoption of the IOT across business and society, serving class-level queries.

5. Scalability to Large IOT Adoption – Serial-level Lookups: S should work in scenarios with a high level adoption of the IOT, serving also serial-level queries.

Performance

The IOTNS S must be able to deliver a performance that is suitable for global use in very heterogeneous applications. This includes:

1. Fast Update Propagation: Information changed by authorized information providers should be propagated fast throughout the system, to avoid stale data.

2. Low Latency: The waiting time for an answer by S to a query shall be short, below one minute to enable nearly real-time operations.[12]

3. Ultra-Low Latency (optional): The waiting time for an answer to a query should be very short, e.g. below a few seconds,[13] to enable real-time or interactive applications with human beings who deem longer waiting times unacceptable.

4. Acceptable Load (Average Node): The network, storage, and processing load of an average node (server) of S must not be too high, to guarantee its correct and fast execution of tasks.

5. Acceptable Load (Special Nodes, Root): The network, storage, and processing load of all special or root nodes (servers) of S must not be too high, to guarantee their correct and fast execution of tasks.[14]

Robustness

S should perform reliably in the face of apparently random errors and attacks common on the Internet.[15]

Again, it must be pointed out that many of those high-level requirements are not yet precise. Depending on particular application scenarios, each high-level requirement should be refined and mapped to several more exact metrics and corresponding

[12] Applications like periodic object inventorying could tolerate much higher latency.
[13] Time range stated by BRIDGE, 2007, pp. 9 [22].
[14] If such nodes exist in the implementation of S.
[15] This requirement will be often grouped together with availability in this thesis.

tolerance intervals, so that their fulfillment can be verified.[16] A similar refinement process can be described in a mathematically rigorous way for security, especially confidentiality requirements.[17] During additional process iterations, additional *time- and domain-specific requirements*[18] – arising from specific application domains – should be combined and reconciled, and again compared to design options.

Given the emerging state of the IOT and its applications today, as well as in other, related fields of research like the structured peer-to peer systems used later in this thesis, we will mostly work with above high-level approximations, focusing on the plausibility of their fulfillment, but give detailed arguments on more precise properties where possible.

The following section will discuss IOTNS security requirements.

2.3 Security Requirements

Someone who places value into an information asset and wants it to be protected is called a *stakeholder* of that asset. This definition allows for a generalization of classical security requirements engineering to the multiple stakeholders of multilateral security.[19] In classic security requirements engineering, for example in parts of the Common Criteria, only one stakeholder is considered, i.e., the owner of a target of evaluation (TOE).[20]

In the following, we use the classical triad of protection goals, which a stakeholder may have with respect to an information asset: availability, integrity, confidentiality.[21] We subsume, for example, anonymity under confidentiality of identity, and authenticity under integrity.[22] Keeping the warning of Gollmann[23] on the subtleties of security definitions in mind, we will nonetheless cite some general definitions here, but in the context of this thesis will restrain from a deeper discussion of the inherent linguistic and semantic complexities, which would, however, be necessary for conducting more formal reasonings on security properties.

- Availability: the property of being accessible and usable upon demand by an authorized entity.[24] Alternatively, formulated by avoidance: the prevention of

[16] For performance metrics cf. Jain, 1991 [97], especially Ch. 3, pp. 30.
[17] Santen, 2006 [178].
[18] Sommerville, 2004, Ch. 6 [188].
[19] See also Gürses and Santen, 2006 [82].
[20] Common Criteria, 2006 [35].
[21] Rannenberg et al., 1999, p. 22 [163]; Gollmann, 2006, pp. 19 [78].
[22] For a detailed discussion of anonymity and related terms cf. Pfitzmann and Hansen, 2008 [156]. For a taxonomy of security vs. dependability concepts, cf. Avizienis et al., 2004 [8].
[23] Gollmann, 2006, pp. 25 [78]: *There is no single definition of security. (...) A lot of time is being spent (and wasted) in trying to define unambiguous notations for security.* Similar Anderson, 2001, pp. 8 [3].
[24] ISO/IEC 13335 [94], also used in ISO/IEC FDIS 27001:2005.

unauthorized withholding of information or resources.[25]

- Integrity: the property of safeguarding the accuracy and completeness of assets. The prevention of unauthorized modification of information.[26]
- Confidentiality: the property that information is not made available or disclosed to unauthorized individuals, entities, or processes. The prevention of unauthorized disclosure of information.[27]

Extending classical security engineering, the concept of *multilateral security* emphasizes the importance of taking the security goals of most or ideally *all* stakeholders into account before designing and building a system, not only of system owners or investors.[28] This becomes especially important for confidentiality requirements of system users.

2.3.1 Availability

The system S, i.e., the IOTNS to be constructed, and its data should be available to authorized users any time they need to access it. We assume this to be a requirement shared by all stakeholders. In particular, S should offer robustness to targeted *(Distributed) Denial-of-Service Attacks* (DDoS); the system should avoid single points of failure, and be able to adjust itself to failures of single components (servers or nodes).

Multipolarity. A specific case of availability concerns the anticipated future role of the IOT as critical IT infrastructure in many countries. Considered as stakeholders of S, those countries will have a high interest that no single one of them controls access to S, or could prevent it from working. This requirement will be discussed in detail in the next chapters.[29]

2.3.2 Integrity

S shall offer *data integrity*, including authenticity of data origin. All unauthorized changes to the data stored in S should be detectable by a client via means integrated into S. S should also prevent *Spamming* and *Pharming Attacks*, which aim to add arbitrary, non-authorized data entries to S. All of those also will be assumed common requirements of all stakeholders.

In addition, there are special cases where data integrity may need to be enforced by *system integrity* in lower-level design steps; for example, to deliver authentic

[25] ITSEC, 1991, after Gollmann, 2006, pp. 19 [78].
[26] ISO/IEC 13335 and ITSEC, 1991.
[27] Ibidem.
[28] Rannenberg et al., 1999, p. 26 [163].
[29] Here whole *nations* can be considered as information security stakeholders.

messages on non-existence of records or during the publishing phase, where the system node that is contacted for publishing needs to be authentic. Data integrity should also include a measure to assess the age of data, as well as provide non-repudiation, i.e., the fact that a provider published exactly this data should be provable to third parties, for example for auditing or legal purposes.[30]

2.3.3 Confidentiality

In classical security engineering, confidentiality requirements usually have been considered only for the information provider (*server*) side of an Internet service, such as confidentiality enforced by access control for the data offered by a Web server. This provider perspective also applies to UC environments and the IOT, but is far from complete. Clients of IOT services are also stakeholders whose security requirements need to be accounted for.[31]

Stakeholder confidentiality requirements[32] on the *client* side of S, however, usually do not only refer to data processed in a system, but also to high-level information (e.g., turnover or lifestyle) inferable from *using the system*, and multiple entities (persons, organizations, competitors, criminals, the public) from whom that information must be kept confidential (counter-stakeholders).[33]

As an example, consider the use of RFID, IOT, and the name service S in a smart home owned by an individual, *Bob Concerned*, and on a shop floor (Fig. 2.2).[34] Bob Concerned practices a lifestyle he wants to keep confidential from others (Fig. 2.2(a)). These others are the counter-stakeholders of his confidentiality requirements, including neighbors, marketing companies, and governments, or other entities who adopt *functional roles* in the IOT (Table 2.1) – for example roles in the EPCglobal Network, see Chapter 3. The Shop has confidentiality goals that have a similar structure to Bob's goals (Fig. 2.2(b)). For example, the Shop produces turnover that it wants to keep confidential from competing shops.

Many of those high-level information assets may be inferable – simply from the observation of queries to S, by using query data analysis and mining from content, location, time, frequency, clusters of queries, changes over time. For example, lifestyle can be inferred by analyzing which item brands are in regular use at Bob's

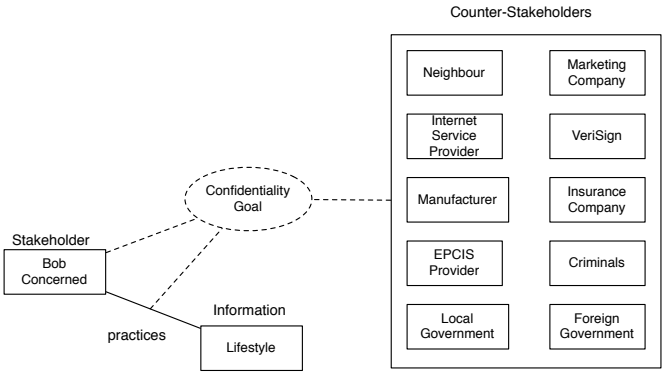
[30] In a sense, document integrity therefore also implies accountability of the provider's action to have published this specific document. Accountability of the *client*, however, is not discussed in this thesis, due to its strong conflict with query confidentiality and client anonymity, which we consider important requirements, and its – in our view – limited relevance to the retrieval of name service data.

[31] This section is summarizing joint work with several researchers: Fabian and Hansen, 2006, [59]; Fabian et al., 2005 [62]; Bauer et al., 2006 [12]; Fabian et al., 2006, [63].

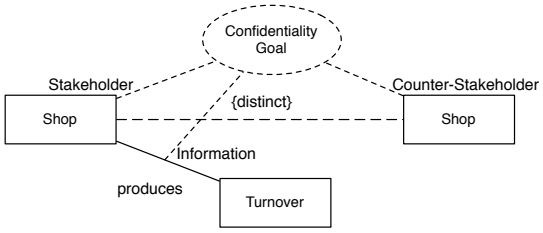
[32] To differentiate these high-level requirements from low-level system specifications, the term *goals* could be used, cf. Fabian et al., 2009 [64].

[33] Gürses et al., 2006 [83]; Gürses and Santen, 2006 [82]; Fabian et al., 2006 [63].

[34] These scenarios were first presented and discussed in joint work [63].



(a) Individual Confidentiality Goals



(b) Shop Confidentiality Goal

Figure 2.2: Example Stakeholders and Confidentiality Goals

home and are creating periodic query patterns to the IOT, or how often new and potentially expensive or cheap items are detected by his RFID readers, while EPCs are resolved to retrieve item information for smart home services. Similarly, the shop's turnover can be inferred by observing periodic inventory queries to the IOT, watching for specific brands, items missing, returns, or new arrivals.

However, Bob Concerned or the Shop may not even be aware of the data traces that they are producing in their smart environments equipped with RFID readers, and across IOT name and information servers, simply by querying and retrieving item-related information.

This situation is typical for UC systems, which in general offer a plethora of low-level data, such as EPC sightings and queries, or seemingly "innocuous" sensor data.[35] Therefore, it will become very difficult for security engineering to state confidentiality requirements for UC in a rigorous way. Not only does the *what* to protect become harder to specify the more concrete a system becomes in the development process, also the *against whom* becomes difficult to state precisely for omnipresent, globally connected machines and environments operated by multiple stakeholders, some of which may not even be known in advance. On the other hand, protection against *all* potential adversaries seems to be impossible to achieve in

[35] Cf. Fabian and Hansen, 2006 [59] for a general discussion. For an illustrative example from sensor network research, see Han et al., 2007 [84].

practice.

Furthermore, mappings between counter-stakeholders and functional system roles are not always clear, in addition to the ever-present possibility of external attacks. Counter-stakeholders and adversaries usually also differ in the degree of background knowledge they have available for data analysis, which could abstractly be described by general conditional probability distributions, but seems nearly impossible to quantify for all adversaries during run-time, or in earlier development stages. To cope with those problems, we adopt a pragmatically-oriented approach and apply a *rule-of-thumb* guideline in this thesis: low-level data in S should be as hard to collect or analyze as possible.[36]

Consequently, while using the IOT, there will be many situations when the EPC belonging to an RFID-tagged item should be regarded as sensitive information – be it in a private context, where people fear to be tracked or have their belongings read by strangers, or in a business context, where product flows constitute valuable business intelligence. The combination of an EPC company identifier and item reference is usually enough to determine the exact kind of object to which it belongs. This information can be used to identify assets of an individual or an organization. If someone happens to wear a rare item or a rare combination of belongings, one could track that person even without knowing the actual serial numbers – we call the latter *Cluster Tracking* in the following. For an overview of possible inferences from query data, see Table 2.2.[37]

In addition to this *supply* side, there are also many entities who have a certain or at least potential *demand* for EPC traces, which will be illustrated in the following section.

Potential Demand for EPC Traces

There are indicators that RFID traces in general and EPC traces in particular will prove to be valuable to many parties.[38] For example, a collection of possible uses is offered in an IBM patent application from as early as 2001:[39]

> *In another embodiment, instead of determining the exact identity of the person, some characteristics such as demographic (e.g., age, race, sex, etc.) may be determined based on certain predetermined statistical information. For example, if items that are carried on the person are highly expensive name brands, e.g., Rolex watch, then the person may be classified in the upper-*

[36] This approach extends the *collection limitation principle* of established privacy guidelines on personal data, e.g. the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, `http://www.oecd.org` (05.2008).

[37] The details of ONS and EPCIS queries will be presented in Chapter 3. Table to be published in Fabian and Günther, 2009 [58].

[38] The following section is based on joint work Bauer et al., 2006 [12].

[39] Hind et al., 2001 [89]; made public by Albrecht and McIntyre, 2005 [1].

Query Data	Inferable Information	Possible Further Analysis
Source IP, Time	Identity and location of the information client. Frequency of objects passing RFID readers.	Who does the query? Where is she located? At what time and how often are items used or processed?
EPC Company Prefix	Item Manufacturer	What general brand is used? High-level consumer preferences. Tracking of very rare brands.
EPCIS DNS Name	Item Manufacturer	(see above)
EPCIS IP	Item Manufacturer	(see above)
EPCIS URL	Item Manufacturer	See above, plus analysis using EPCIS directory structure.
Unencrypted EPCIS Reply	Item information related to this EPC	Varies with specific content.
Partial EPC (Company Prefix + Item Reference)	Item Manufacturer Object class	Exact item category of this brand. Detailed consumer preferences. Rare item or Cluster Tracking.
ONS Query	Item Manufacturer Object class	Exact item category of this brand. Detailed consumer preferences. Rare item or Cluster Tracking.
EPC (Company Prefix + Item Reference + Serial Number)	Unique Object Identity	Which unique item within this kind and brand? Detailed consumer preferences. and exact buying behavior. Item or Cluster Tracking. Identity of object owner or holder. Social or business networks.

Table 2.2: Inference Examples

> *middle class income bracket. In another example, if the items that are carried on the person are "female" items typically associated with women, e.g., a purse, scarf, panty hose, then the gender can be determined as female. [...]* (p. 2)
>
> *When a person enters a retail store, a shopping mall, an airport, a train station, a train, or any location where a person can roam, a RFID-Tag scanner located therein scans all identifiable RFID-Tags carried on the person [...]* (p. 3)

This patent application gives anecdotal evidence that some experts did foresee the potential usefulness of gathering quality information through RFID traces.[40] In the following, a more detailed view on the motivations of potential trace consumers is presented.

Companies. There are many reasons for the private sector to develop a substantial demand for EPC traces, possibly to such an extent that even companies that gather their own supply of traces will have reason to buy from or pool with other companies for data completion, integration, and refreshment.

The first motivation is the enhanced potential for *personalization and direct marketing*. There are empirically verified economic benefits for companies to personalize

[40] IBM has also entered the market for RFID privacy solutions later, cf. Karjoth and Moskowitz, 2005 [107].

their goods or services.[41] This has influenced industry best-practices.[42] Personalization and recommendation systems especially will be highly pertinent to in-shop or home RFID applications (B2C), and will also increase demand for trace data in E-Commerce. Some benefits of personalization[43] are the ability to turn casual browsers into buyers, the potential of cross-sells by recommending matching items to something already owned by the customer, increased customer loyalty, and better customer relationship management. In addition, customers will be a much more convenient target for product placement and direct marketing strategies.[44]

Personalized insurances[45] will stimulate a huge demand for traces by insurance companies to study a person's whereabouts, movements, and consumption habits.[46] Traces will also enhance the effectiveness of credit scoring by providing detailed insights into the subjects' possessions and income.

The second motivation is *price discrimination.* Andrew Odlyzko has convincingly identified price discrimination as an important driver for privacy erosion on the Internet.[47] To maximize profit for a service provider or merchant, a customer should ideally pay the maximum amount that is acceptable to her. In order to charge different customers different prices for the same service or good, data is needed to estimate their *willingness to pay.*[48] Data generated through personalization of shopping sites, click tracing, and other measures used on the Internet are conducive to such an analysis.[49] EPC traces will be a new source of relevant information, pertaining to the physical world.

Third, there is enhanced potential for *business intelligence and industrial espionage.* Players in many industries will be tempted, if not actively interested, in the possibility of inspecting a competitor's supply chain or in lists of items or persons who enter their buildings. Also, less aggressive business intelligence can make use of EPC traces as well, for example to investigate trade relationships by analyzing physical flows of goods using their virtual footprints.

Governments. For governmental agencies it will often be more convenient to accumulate raw or personalized traces from private companies, rather than to invest into additional reader infrastructures that cover sufficient area for permanent

[41] Pine II et al., 1993 [157].

[42] E.g., Peppers et al., 1999 [153].

[43] Schafer et al., 1999 [179].

[44] Cf. Section 3.4.4 for recent developments on massive Web traffic collection and analysis for advertising purposes.

[45] An example today is the *Pay as you drive* insurance, URL: `http://www.norwichunion.com/pay-as-you-drive/` (03.2008).

[46] Bohn et al., 2004 [18].

[47] Odlyzko, 2003 [142].

[48] Price discrimination, if it becomes public, is not without risk for the service provider's image. However, the more complex and personalized a service becomes, the harder price discrimination may be to detect.

[49] Cranor, 2003 [38].

surveillance. Today in the US, government agencies often buy personal data from profile brokers like ChoicePoint.[50] This trend could extend to EPC traces, enhancing the information gathered by public readers that are installed by the state, e.g., for ticketing, traffic monitoring, billing, and building security.

Some of the potential utilities of EPC traces for governments are: *Customs and Tax Collection.* Ownership of goods, their transfer and movement patterns are very interesting to customs authorities that could now track imported and exported goods. Likewise, tax collection for luxury items will be made easier by tracking items and their owners. Simply the threat of this possibility may be expected to reduce delicts and misdemeanors.

Disaster Recovery or Prevention. Furthermore, ideas such as supporting civil disaster recovery or prevention plans through new technologies, e.g., in case of epidemics, may be possible once trace databases have become sufficiently large and accurate.

Law Enforcement. The police will have a high interest in traces, as they will prove extraordinarily useful in forensics and perhaps even crime prevention. Monitoring and remote surveillance of criminals or suspects will be facilitated. A similar argument holds for *Intelligence Agencies.* Even if they may already have access to equivalent information, traces could be used as confirming evidence to reduce uncertainties. Live traces could support other forms of surveillance, and social (e.g., terrorist) networks could be analyzed more easily. Nevertheless, the challenge of false positives and false negatives will have to be tackled even more seriously with the increase of EPC traces.

Individuals and Researchers. Individuals will also be interested in EPC traces. This may be to quench natural human curiosity, or for more sinister activities such as blackmailing or spying on neighbors, relatives, or co-workers. On the other hand, applications for child care as well as care for elderly persons could make use of Ubiquitous Computing and IOT data, and may be appreciated by their users for improving their quality of life. Finally, there could be a substantial demand in EPC traces to support scientific research. Examples include economics (e.g., improving research on trade), medical research in epidemics, migration and mobility research, and social sciences in general.

To conclude, traces generated by a client of the IOT and a corresponding name service S will be valuable for multiple parties, whose purposes may conflict with the clients' confidentiality goals. Therefore, query confidentiality is an important issue for the IOT and its name services.

[50] References gathered by EPIC: `http://epic.org/privacy/choicepoint/` (03.2008).

Confidentiality Requirements

Reflecting upon the previous discussion, the following list tries to summarize the most urgent confidentiality requirements S needs to fulfill in the various application domains of an IOT. As with functional and other requirements stated in Section 2.2, these are high-level requirements that will need to be refined throughout application domains and development iterations (if possible, cf. the discussion in Section 2.3.3).

1. Confidentiality of Address Data: The provider should have the option to implement access control to the OIS (EPCIS) address documents he publishes to S, and to keep it confidential during transmission. A client may share this requirement due to his intention to keep the query content confidential.

2. Confidentiality of Object Data: If actual object data is stored in S, information providers must be able to implement access control according to their own policies, and to keep the data confidential during transmission. A client may share this requirement due to her requirement to keep the query content confidential, or because the object data contains otherwise sensitive information. Other stakeholders, perhaps not even participating in the IOT themselves, could have confidentiality requirements w.r.t. object data as well.

3. Confidentiality of Provider Identity (Service Anonymity): The information provider may want to hide the fact he is offering information for a specific OID from particular counter-stakeholders and adversaries. This is but a hypothetical requirement, which could become necessary in general Discovery Service scenarios.[51]

4. Confidentiality of Client Identity (Client Anonymity): The identity of the querying client should remain confidential from specific counterstakeholders and adversaries.[52]

5. Confidentiality of Query Content: The content of the query, especially the OID and its parts, should remain confidential from specific counter-stakeholders and adversaries.

6. Query Confidentiality (QC) – a specific definition used in this thesis: In the following chapters, we often refer to a combination of client anonymity and confidentiality of query content as *query confidentiality*.

   The requirement of query confidentiality shall be satisfied if not both elements of a pair (Identity, Query content) become known to an adversary, for example

[51] In this thesis, we also do not increase this requirement's granularity (Anonymity, Confidentiality of Provider Location, Unobservability).

[52] Using the definition of anonymity given in Pfitzmann and Hansen, 2008 [156], the anonymity set in this requirement would be the set of all users of S.

as (IP, EPC) tuple.[53] This requirement concerns the *relation* of client ID and query content, as depicted in Fig. 2.2, and may be of higher practical relevance to stakeholders than both aforementioned requirements.[54]

Ideally, QC could be achieved by the *conjunction* of anonymity *and* confidentiality of query content – the most robust way of fulfillment in the face of unknown adversary background knowledge, *strong QC*. But QC could also be satisfied by their *disjunction*, that is, by keeping at least one element of the pair confidential – with a higher risk of potential mutual inference between the tuple elements, *weak QC*.

7. Confidentiality of Client Location: The physical location of the querying client should remain confidential against specific counter-stakeholders and adversaries.

8. Client Unobservability: The whole query process should be unobservable by specific counter-stakeholders and adversaries. Unobservability vs. all possible counter-stakeholders, however, is a very demanding requirement, nearly impossible to fulfill in practice.

While designing and building real-world systems, especially on a global scale with multiple stakeholders, conflicts[55] between functional and security requirements, or between security requirements of different stakeholders, are nearly inevitable. One example is the potential conflict between provider data confidentiality vs. availability, where an increase of copies to enhance availability may increase the risk of confidentiality breaches. Those conflicts need to be resolved by trade-offs, that is, by weakening some the conflicting requirements in further development phases, or by mechanisms that are able to reconcile them. In general, the more concrete a system becomes, the less absolute its security guarantees become because of additional attack paths that appear.[56]

2.4 Requirements Overview

This section summarizes the preceding discussions on IOT name service requirements. Note again that this list is necessarily incomplete, because during additional design and security analysis iterations new requirements may appear, existing requirements may be changed and refined, or get weakened due to trade-offs in the reconciliation process with other requirements. On the other hand, there are many different ways to logically structure security requirements (see Section 2.3), therefore

[53] This could also be described as an anonymity requirement, in which the anonymity set would be the set of all users of S who query for this specific EPC.

[54] Cf. joint work Fabian et al., 2006 [63].

[55] Also called requirements *interactions*, cf. Sommerville, 2004 [188].

[56] For a systematic and formal treatment of that phenomenon see Santen, 2006 [178].

some requirements may be only indirectly referred to here.Table 2.3 presents the requirements identified so far, using the following notations: *Category* indicates *F* for functional, *NF* for non-functional, and out of those in particular *SEC* for security requirements. The field *Stakeholders* indicates the subset of IOT stakeholders who

Category	**Stakeholders**	**Requirement**	**Counter-SH**
F	SH	Membership and Authorization	–
F	SH	Flexible OID Support	–
F	SH	Single Publisher for specific OID	–
F	SH	Publishing	–
F	SH	Querying	–
F	SH	Class-level Addresses	–
F	SH	Serial-level Addresses	–
F	SH	Object Information	–
NF	SH	High Node Count	–
NF	SH	High Client Count	–
NF	SH	Medium IOT Adoption	–
NF	SH	Large IOT Adoption – Class-level	–
NF	SH	Large IOT Adoption – Serial-level	–
NF	SH	Robustness (Random Error)	–
NF	SH	Fast Update Propagation	–
NF	SH	Low Latency	–
NF	SH	Ultra-Low Latency	–
NF	SH	Acceptable Load (Leaf Server / DHT Node)	–
NF	SH	Acceptable Load (Root, TLD)	–
SEC	SH	Availability	ADV
SEC	$\subseteq$ SH	Multipolarity	$\subseteq$ (SH $\cup$ ADV)
SEC	SH	Integrity of System	ADV
SEC	SH	Integrity of Data	ADV
SEC	$\subseteq$ Providers, Clients	Confidentiality of Address Data	$\subseteq$ (SH $\cup$ ADV)
SEC	Providers, Clients, Others	Confidentiality of Object Information	$\subseteq$ (SH $\cup$ ADV)
SEC	$\subseteq$ Providers	Confidentiality of Provider Identity	$\subseteq$ (SH $\cup$ ADV)
SEC	Clients	Confidentiality of Client Identity	$\subseteq$ (SH $\cup$ ADV)
SEC	Clients	Confidentiality of Query Content	$\subseteq$ (SH $\cup$ ADV)
SEC	Clients	Query Confidentiality	$\subseteq$ (SH $\cup$ ADV)
SEC	$\subseteq$ Clients	Confidentiality of Client Location	$\subseteq$ (SH $\cup$ ADV)
SEC	$\subseteq$ Clients	Client Unobservability	$\subseteq$ (SH $\cup$ ADV)

Table 2.3: High-level Requirements Summary

may state this requirement, *SH* stands for the set of all IOT stakeholders, $\subseteq$ for subsets.

Stakeholders may be parties who adopt functional roles in the IOT – such as object address or object information *Providers* and *Clients*, as well as IOT *Infrastructure Providers* like EPCglobal – but the stakeholder set is not limited to those. It includes in general all entities regarding the information processed in the IOT as an asset that they place value on, for example also people whose EPCs get processed in the IOT, or countries who have a political interest in multipolarity of the IOT. As noted earlier, the extent of this group may change over time and could face enormous growth rates due to the possible diffusion and extension of the IOT to Ubiquitous Computing applications – therefore, the membership of SH will be hard to determine in advance. This emphasizes again the necessity for future iterations of the security requirements elicitation process.

Counter-SH indicates the counter-stakeholders of this requirement, in the sense of *someone else* who *potentially* places value on the asset of a stakeholder, or places value on the violation of a security requirement. In our definition, a counter-stakeholder does not need to be a stakeholder of the IOT itself. Therefore, *Counter-SH* may include IOT stakeholders including all representatives of functional roles of the IOT, but also external adversaries. *ADV*, the set of *adversaries*, describes the entities that place an *actual* value in violating security requirements, and are actively trying to achieve this purpose.[57]

In practice, several counter-stakeholders $c \in SH$ and $d \in ADV$ might be colluding, or a dependency could exist between a counter-stakeholder who is delegating the active task to an adversary. *ADV* also includes non-human entities that act randomly without directed purpose, such as Internet Worms, or that have only a latent interest in a stakeholder's IOT-information asset, such as general purpose malware.[58]

Table 2.3 could serve as a guideline for future iterations of IOT and IOTNS requirements elicitation, e.g., when specific stakeholders are interviewed about their particular security needs.

2.5 Summary

This chapter introduced RFID, EPC, and the *Internet of Things* (IOT), as well as two main application areas. The role of name services for the IOT was defined, and high-level requirements for those name services were investigated. From the set of security requirements, client confidentiality requirements in particular were shown to be an important, complex, but so far neglected problem in the face of many interested parties, which include other stakeholders of the IOT and external adversaries.

The following chapter presents the most influential architecture proposal for the IOT, the EPCglobal Network and its name service ONS, and discusses its security shortcomings, contrasting them to the requirements presented here.

[57] This sense of activity encompasses both *active* and *passive* adversaries in the sense of cryptography or network security engineering, adversaries who are participating in protocols, or are just passively eavesdropping.

[58] Reflecting on these definitions, Table 2.3 actually depicts possible *requirement schemes* for compact readability, which will be refined – using Table 2.2 – and instantiated with respect to specific members of the set *Counter-SH* in later chapters of this thesis. Using the terminology of [64], the lower part of the table can be regarded as containing schemes of high-level security *goals*, which could be refined to *requirements* and detailed *specifications*.

Chapter 3

ONS Security Challenges

3.1 Introduction

In this chapter, we will introduce the most influential proposal for an IOT infrastructure, the EPCglobal Network, and its main name service.

We will present a security analysis of this most influential IOTNS standard, the *Object Naming Service* (ONS), and discuss its major security shortcomings, which are already recognizable today in an early state of design and deployment. This analysis is based on the first published security analysis of the ONS, Fabian et al., 2005 [62], and on the discussion of the confidentiality challenges of EPCglobal Network as a whole, to be published in Fabian and Günther, 2009 [58].

Related work includes a survey of classical security measures for the EPCglobal Network conducted by the AutoID labs, Konidala et al., 2006, [113], which however does not discuss client confidentiality requirements. Risks to backends from malicious RFID tags have been described by Rieback et al., 2006 [171], also one of the first papers changing the point of view from tag-reader security to backend systems.

RFID middleware security is also discussed in Song et al., 2005 [190], and Song and Kim, 2006 [189]. The BRIDGE project investigated security for Discovery Services: BRIDGE, 2007 [23].

The chapter has the following structure. First, the EPCglobal Network – the reference IOT architecture – is described. Then ONS is presented in detail, including a discussion of its DNS foundations. This is followed by the security analysis of ONS, where special emphasis is placed on risks to client confidentiality.

3.2 EPCglobal Network

EPCglobal,[1] originating from the Auto-ID labs of MIT, the former EAN International and Uniform Code Council (both now GS1), is a consortium that places its focus on developing and establishing global standards for RFID, EPC, and the EPCglobal Network.[2] According to their intention, information about an object should in general not be stored on its RFID tag itself, but instead be supplied by distributed servers on the Internet.[3] By using the EPC and the help of name services like the *Object Naming Service* (ONS)[4] and *EPCIS Discovery Services*,[5] it will be possible to locate EPC Information Services (EPCIS), which are remotely accessible data collections about the particular object.[6]

One of the advantages the EPCglobal Network offers is to let many parties – manufacturers, suppliers, shops, or after-sale service providers – dynamically register any kind of EPCIS for the objects they are concerned with, thereby creating an open way to exchange product related information. By improving the information flow, as objects pass from suppliers to manufacturers, distributors, retail stores, and customers, the EPCglobal Network aims to facilitate cooperation within supply chains and thus to make them more efficient.[7]

Once established, it could also be used to support a wide range of applications in the area of Ubiquitous Computing (UC). An example is the *smart home*, in which "intelligent" cupboards and refrigerators could be realized using RFID technology. By scanning the RFID tags on objects and using the EPCglobal Network for information retrieval, such devices can identify their current content and offer new services such as food counseling or automated replenishing of goods.[8]

As a result of this potentially broadened use of the EPCglobal Network, its security context will change from closed supply chains to the rather open environments of UC – like the security context of the Internet and the Web was changed by moving from relatively closed groups of fellow researchers to the global environment it represents today.

The main components, i.e., interfaces and official functional roles in the EPCglobal Network are depicted in Fig. 3.1, an illustration taken from the official documentation.[9]

Normal participants of the EPCglobal Network are called *EPCglobal Subscribers*.

[1] URL: `http://www.epcglobalinc.org` (03.2008).

[2] Also simply called *EPC Network* in some documents and in general use. We use the terms as defined in EPCglobal, 2007 [53].

[3] Ibidem.

[4] Sometimes called Object *Name* Service in official documents.

[5] Also called *EPC* Discovery Services.

[6] Harrison, 2004 [86].

[7] Cf. Section 1.5.

[8] Cf. Section 1.6.

[9] EPCglobal, 2007, p. 27 [53].

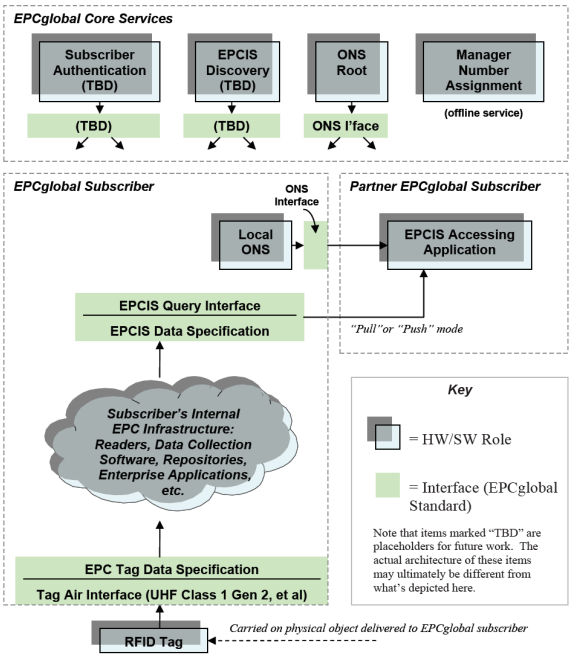


Figure 3.1: EPCglobal Network Roles and Interfaces (Source: EPCglobal)

They capture EPCs and data from RFID tags on objects they receive via standardized *Tag Air Interfaces* by using their RFID-reader infrastructure. The data and EPCs are passed to the internal Intranet and EPC infrastructure, which besides RFID readers includes higher layer collection and aggregation software, repositories, as well as enterprise applications. This EPC infrastructure can itself offer and query EPC Information Services (EPCIS), both locally and remotely to partner EPCglobal Subscribers, enabling the exchange of object and event data. A *Local ONS* server is responsible for offering EPCIS-address information to remote partners.

EPCglobal itself offers *Core Services*, like the offline *Manager Number Assignment* for assigning and managing the EPC Manager part of EPCs, and online *Subscriber Authentication*, and *EPCIS Discovery*.[10] In addition, EPCglobal is responsible for the *ONS Root*, whose practical operation has been outsourced to the company VeriSign.[11]

In order to locate dynamically registered EPCIS globally, a static list or a single server would lead to out-of-date information and scalability problems.[12] The EPCglobal Network therefore includes central name or look-up services called EPCIS Discovery Services and Object Naming Service (ONS).[13] Each time someone requests information about a particular object – information not already present in local caches, or "stale", that is, marked as out of date – these services are queried

[10] Both yet to be published at the time of this writing.
[11] EPCglobal, 2005 [50].
[12] Uo et al., 2004 [202].
[13] EPCglobal, 2007 [53].

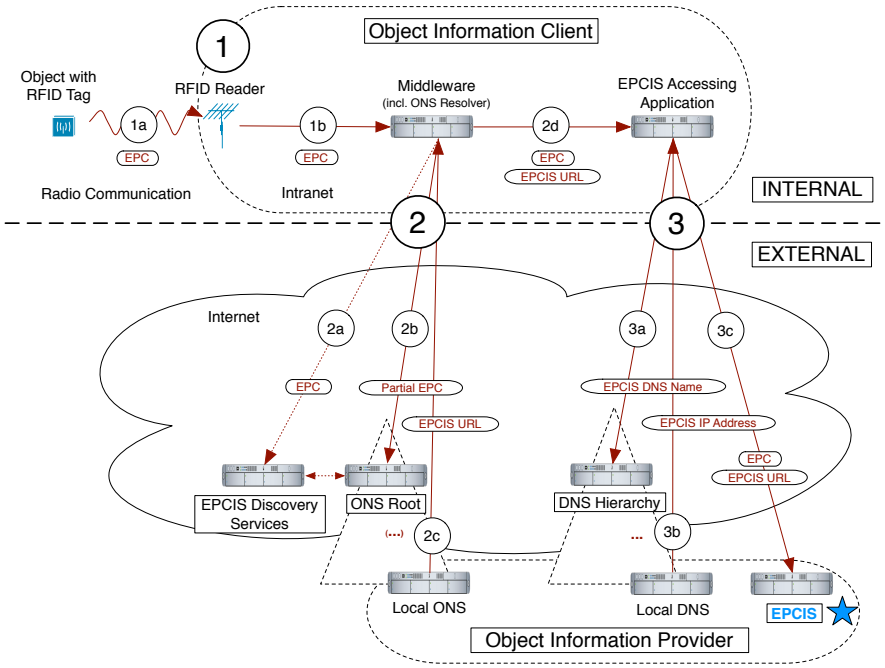


Figure 3.2: EPCglobal Network Communication Flow

for a recent list of relevant EPCIS. After retrieving this list, the requestor directly contacts the EPC Information Services in which she is interested. Thus, object information retrieval in the EPCglobal Network generally consists of three main phases (cf. Fig. 3.2):

1. RFID Tag-to-Reader and Intranet Communication: An RFID reader reads an EPC from an RFID tag via wireless communication (1a). This EPC is transmitted to a middleware layer for further processing (1b).

2. EPCIS Discovery and ONS: This phase will involve EPCIS Discovery Services that are not specified yet (2a).[14] The middleware queries ONS for Uniform Resource Locators (URLs) of corresponding information sources (mostly EPCIS) (2b).[15] The final answer (2c) from the Local ONS of an information provider is handed over to the application (2d).

3. EPCIS Access: The application needs to resolve the EPCIS DNS names (3a, b) delivered by ONS, and finally contacts the relevant EPCIS directly to retrieve the object information (3c). This procedure will in most cases not be conducted manually, but in an automated fashion, e.g. by the use of Web services.[16]

[14] Ibidem.
[15] Mealling, 2005 [129].
[16] Leong et al., 2004 [119].

Since ONS is the central IOTNS of the EPCglobal Network, we focus on its security issues in the following sections.[17]

3.3 Object Naming Service (ONS)

For ONS, a hierarchical, tree-like architecture has been proposed by EPCglobal.[18] The ONS protocol is identical to the protocol used by the Domain Name System (DNS). The ONS Root is the central root of this tree. Further delegation works as in DNS, and information providers itself will deploy authoritative ONS servers – for their EPC ranges – that point to their actual EPCIS.

This architecture and protocol choice will have a deep impact on the reliability, security, and privacy of the involved stakeholders and their business processes, especially for information clients, as will be discussed after the technical inheritance of DNS has been described in the next section.

3.3.1 ONS Foundation: DNS

From a technical point of view, ONS is a subsystem of the Domain Name System (DNS), whose history, architecture and protocols are described in Liu and Albitz, 2006 [122], and are codified in many Requests-for-Comments (RFCs).[19]

The main design idea of ONS is to first encode the EPC into a syntactically correct domain name, then to use the existing DNS infrastructure to query for additional information. This procedure makes use of the Naming Authority Pointer (NAPTR) DNS record,[20] which is also used with other Internet applications, for example the Session Initiation Protocol (SIP) for Voice-over-IP (VoIP) to map phone numbers into corresponding Uniform Resource Identifiers (URI).

For a discussion of the DNS security heritage to ONS later in this chapter, in the following sections a short summary of the inner workings of DNS is given, discussing names, architecture, and protocol.

3.3.2 DNS Names and Architecture

The basic function of the DNS is that of an Internet name service: the resolution of human-memorizable, alpha-numerical hostnames into the corresponding purely numerical Internet Protocol (IP) addresses used for datagram routing. At an early stage of the Internet, the ARPANET, name resolution was performed by referring

[17] The initial analysis has been given in Fabian et al., 2005 [62].
[18] Mealling , 2005 [129].
[19] Collected for example by Salamon [177].
[20] RFC 2915, Mealling and Daniel, 2000 [130].

to a flat text file that stored mappings between the hostnames and the IP addresses (`hosts` file).[21] Obviously, maintaining and synchronizing copies of the `hosts` file on all computers connected to ARPANET was extremely inefficient.

To address this issue, the name resolution protocol was updated to introduce a central distribution of the master `hosts` file via an online service maintained by the Network Information Center. This architecture worked successfully for about a decade. However, the rapid growth of the Internet rendered this centralized approach impractical. The increasing number of changes introduced to the `hosts` file and its growing size required hosts to regularly download large volumes of data and often led to propagation of network-wide errors.

As a reaction, shortly after deployment of TCP/IP, the new Domain Name System (DNS) was introduced.[22] This DNS still serves as the foundation of the Internet name resolution system today. A hostname now has a compound structure and consists of a number of labels separated by dots, e.g. `www.example.com.` – the final dot is often omitted. The labels specify corresponding domains: the empty string next to the rightmost dot corresponds to the *root domain*, the next label to the left to the *top-level domain* (TLD), followed by the *second-level domain* (SLD) and so forth.

The resolution of the hostname into the corresponding IP address is carried out by a tree-like hierarchy of DNS name servers. Each node of the hierarchy consists of DNS nameservers that store a list of *resource records* (RRs) mapping domain names into IP addresses of Internet sites belonging to a *zone* for which the DNS servers are authoritative. Alternatively, in case of zone delegation, IP addresses of DNS servers located at the lower levels of the hierarchy are returned. The resolution of a hostname is performed by subsequently resolving domains of the hostname from right to left, thereby traversing the hierarchy of the DNS nameservers until the corresponding IP address is obtained.

In addition to name-to-IP resolution used by nearly every Internet application today, there are several other established and future uses of DNS, such as inverse queries (IP-to-name), queries for mail server addresses (using MX RRs), DNS use for storing VoIP phone numbers,[23] key distribution,[24] and even for stating communication security requirements.[25] ONS will place additional DNS burden on top of the load created by all those applications.

[21] Liu and Albitz, 2006, p. 3–4 [122].
[22] First and central RFCs include RFC 1034 [132] and 1035 [133] (Mockapetris, 1987).
[23] ENUM, RFC 3761.
[24] Especially for IPsec, cf. RFC 4025.
[25] Ozment et al., 2006 [147].

3.3.3 DNS Protocol

The DNS protocol is part of the application layer of the TCP/IP hierarchy.[26] In general, it uses the User Datagram Protocol (UDP) with server port 53 as transport layer protocol for queries and responses. Mainly out of historical reasons, DNS uses the Transmission Control Protocol (TCP) for responses larger than 512 bytes, as well as for higher reliability of zone transfers between DNS servers. An exception is the use of so-called Extension Mechanisms for DNS,[27] which allow larger DNS payloads to be transported via UDP, and is important for transferring signatures for DNS Security Extensions (DNSSEC).

To match incoming responses with previous queries, DNS uses a 16 bit query identifier located in the DNS header. In addition, the header carries multiple status bits indicating query or response, authoritative answer, response truncation, and the desire for – respectively, availability of – recursive query tasks for the name server.[28] The actual query or answer DNS RRs, as well as possible additional information, follow after the header in specific section of a DNS packet. To reduce the message size due to the classical 512 byte limit, a compression and pointer scheme is used to avoid the repetition of names,[29] which however increases the parsing complexity for human eye and DNS software, and has been the cause of implementation errors in the past.

In the following, the inner workings of the ONS resolution process and its use of DNS are described.

3.3.4 ONS Resolution Process

The ONS resolution process is described in Mealling, 2005 [129], as well as by an earlier article.[30] For a schematic view of the communication procedure, cf. Fig. 3.3.

After an RFID reader has received an EPC in binary form, it forwards it to some local middleware system. To retrieve the list of relevant EPCIS servers for this particular object, the middleware system converts the EPC to its URI form (e.g. `urn:epc:id:sgtin:809453.1734.108265`).[31] Then this is handed over to the local ONS resolver, which in turn translates the URI form into a domain name (e.g. `1734.809453.sgtin.id.onsepc.com`) by following a well-defined procedure.[32] This name belongs to a subdomain of the domain `onsepc.com`, which is reserved for ONS use.

[26] Stevens, 1994 pp. 187 [194].
[27] EDNS0, RFC 2671.
[28] Mockapetris, 1987 pp. 25 [133].
[29] Ibidem, pp. 29 [133].
[30] Uo et al., 2004 [202]
[31] For different EPC representations cf. EPCglobal, 2007 [51].
[32] Mealling, 2005, Section 5 [129].

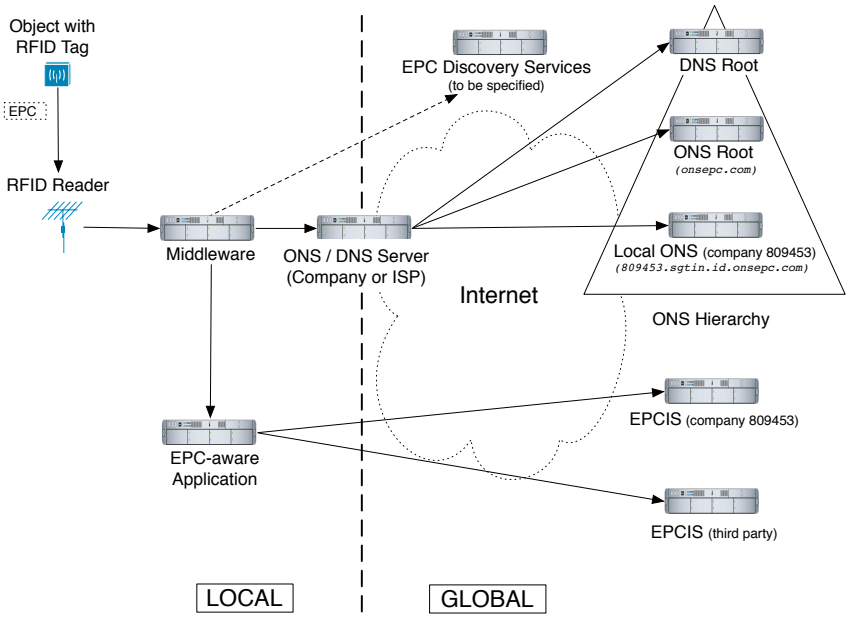


Figure 3.3: ONS Resolution

The current ONS specification states that the serial part (item level, in the example: 108265) of the EPC, which differentiates between objects of the same kind and brand, should not be encoded as of now, but it leaves room for such a possibility:[33]

> *The ability to specify an ONS query at the serial number level of granularity as well as the architectural and economic impacts of that capability is an open issue that will be addressed in subsequent versions of this document. Its lack of mention here should not be construed as making that behavior legal or illegal.*

This newly created domain name is now queried for by using the common DNS protocol, possibly involving a recursive query to a local DNS or service provider DNS server that then queries iteratively accross the Internet.[34]

In addition to this primary ONS use of the DNS, note a secondary dependency on the existing DNS hierarchy: The names stored in ONS records and returned by the ONS query process will again have to be resolved into IP addresses – using the standard DNS hierarchy – to receive the IP addresses of the EPCIS servers, cf. (3a, b) in Fig. 3.2.

We turn now to the actual discussion of ONS security issues, especially with respect to the DNS heritage, which becomes critical in this new application domain.

[33] Ibidem, Section 3.2.1 [129].

[34] With *iterative* queries, the client itself queries server by server in the hierarchy, while *recursive* queries demand that someone else does the work and should just deliver the result back, see Liu and Albitz, 2006, pp. 29 [122]. Similar concepts exist in DHT systems, cf. Section 5.4.2.

3.4 ONS Security Analysis

DNS is an old and central Internet service with a long history of security and configuration issues in the protocol itself and in particular implementations.[35] Various vulnerabilities and attacks can be listed by consulting established security sites as CERT,[36] SecurityFocus,[37] and the SANS Institute's *Top 20 List of Internet Security Vulnerabilities*.[38]

A corresponding Request-for-Comments, RFC 3833 *Threat Analysis of the Domain Name System*,[39] was published quite late after two decades of DNS use, though many of its security problems have been identified before. Some of the main threats discussed are: packet interception, i.e., manipulating IP packets carrying DNS information, query prediction by manipulating the query and answer schemes of the DNS protocol, cache poisoning by injecting manipulated information into DNS caches, betrayal by trusted servers controlled by an attacker, and denial of service, a threat to every Internet service – but DNS itself might be used as an amplifier to attack third parties.[40]

Besides bugs in the code, the fundamental reason for most of these vulnerabilities is the fact that even though DNS is a central and highly exposed service by definition, it has – in its original and widely deployed form – no way of authenticating a client, the server, nor the information that is provided. In addition, DNS uses a clear text protocol, as do most of the early Internet protocols.[41]

These DNS weaknesses directly transfer to ONS. In the following sections, a discussion on ONS availability, integrity, and confidentiality risks is given.

3.4.1 ONS Availability

ONS will constitute a service highly exposed to attacks from the Internet, if only due to its necessary widespread accessibility. A particular threat is *Denial of Service* (DoS), which abuses system and network resources to make the service unavailable or unusably slow for legitimate users.[42]

This could include *Distributed Denial-of-Service* (DDoS) attacks overwhelming a particular server or its network connection by issuing countless and intense queries,

[35] Vixie, 1995 [204]; Pappas et al., 2004 [149]; Kaminsky, 2006 [105].
[36] CERT Search URL: `http://search.cert.org/` (04.2008).
[37] URL: `http://www.securityfocus.com/` (04.2008).
[38] URL: `http://www.sans.org/top20/` (04.2008).
[39] Atkins and Austein, 2004 [7].
[40] DNS Amplification Attacks, ibidem, p. 7 [7]. These attacks, often conducted via Botnets, use IP and DNS spoofing to let DNS servers flood a victim by unsolicited DNS responses, exploiting an asymmetry in DNS query and response size. For mitigation attempts cf. Kambourakis et al., 2007 [104].
[41] E.g., IP itself, UDP or TCP, or application layer protocols like HTTP, SMTP, POP3.
[42] Needham, 1993 [138]; Shirey, 2000, p. 55 [184]; Cheung, 2006 [30].

e.g., by the use of *zombie networks*, Botnets or so-called Puppetnets, i.e., hosts controlled by browser-based malware.[43] DoS attacks can also use more sophisticated methods, e.g. targeted exploits that shut down the DNS server software or the operating system.

Though distributed, DNS suffers from limited redundancy in practical implementations. Authoritative name servers for any given zone should be redundant according to RFC 1034.[44] Recent studies on real implementations, however, show that for a non-insignificant part of the global name-space this requirement does not hold.[45] Name servers storing the same information for a given zone are often few and not redundantly placed with respect to geographical location and IP subnets, and often reside inside of the same Autonomous System (AS). There are many servers that have single distinct routing bottlenecks on paths to reach them – from every place in the world.

The small number of servers for a given zone information, and their limited redundancy creates single points or small areas of failure. Those are also attractive targets for Denial-of-Service Attacks – not only at the DNS root, which is currently run by fewer than 150 servers[46] and has been attacked with some, but so far moderate, success before.[47]

Failure of the root, though, would – after some time to account for caching – imply failure of the whole system, not only of some of its subtrees. Root and top-level domain (TLD) servers, as well as name servers for domains that rise in popularity (*flash crowds*, for example the famous Slashdot effect) suffer from strong load imbalance induced by the architecture. Omnipresent DNS caching, on the other hand, reduces flexibility and the speed of update propagation. Studies also show the significance of human configuration errors that slow down the resolution process or even cause it to fail.[48] Part of the problem is the complexity of the DNS delegation process, which is based on cooperation across different organizations.

Therefore an integration of the EPCglobal Network – with ONS as proposed – into core business processes could leave even formerly non-IT related companies dependable on the availability of Internet services. This will most probably increase overall business risk.

Unipolarity. Another facet of DNS politics relevant to ONS availability is a rather global political problem. Who should control and operate the root and TLD servers, and the name space as a whole? To let a single company, in addition to its major role in the DNS root and CA services, take control of the ONS root may

[43] Geer, 2005 [74]; Lam et al., 2006 [116]; see also Provos et al., 2007 [160].
[44] Mockapetris, 1987 [132].
[45] Ramasubramanian and Sirer, 2004 [161].
[46] Gibbard, 2007 [76].
[47] For example, in 2002, 2006, and 2007, cf. Lawton, 2007 [118]; ICANN, 2007 [92].
[48] Pappas et al., 2004 [149]; Wessels, 2004 [214].

hinder international acceptance of the system as a whole. For a more in-depth discussion, see Section 4.2.

3.4.2 ONS Integrity

Integrity in the ONS context refers to the correctness and completeness of the returned information; that is, in general, addresses of EPC Information Services corresponding to the queried EPC. An attacker controlling intermediate DNS servers or launching successful *Man-in-the-Middle* (MITM) attacks[49] on the DNS communication could forge the returned list of URIs and include, for example, a malware-hosting server under her control.

DNS spoofing attacks are quite easily possible because there are no widely deployed integrity-preserving measures in the DNS protocol, for UDP, or the IP layer. The problem of predictable DNS packet IDs that allow forging by a MITM was already discussed in 1989, but has remained a major issues since then.[50] Combined with those, or independently, *Cache Poisoning* attacks pollute the records stored by resolvers and non-authoritative name servers.[51]

Massive, real-world – mostly malware-induced – client-side cache poisoning or modification of resolver `hosts` files to redirect Web traffic to malicious server farms is also known as *Pharming*. Another local attack vector is to modify client routers that are protected only by default passwords to change the DNS resolution of all local clients.[52]

For cyber-crime, often very transient association between domain names and IP addresses are used (*Fast Flux Networks*), which make criminal sites hard to track and shut down.[53] Not least, there is the vast history of implementation errors and bugs in DNS server and client software, which will not be different for ONS. Exploits continue to be produced to conquer unpatched servers and control the information they contain.

If there are no sufficient authentication measures for the EPCIS in place, the attacker could deliver forged information about this particular or other related EPCs from a similar domain. The corresponding risks will be specific to the application: If the query was initiated by a smart refrigerator to order matching ingredients for a cooking recipe, this could result in spoiled meals; if the query was issued by a smart medicine cabinet – as a precursor to an even smarter home medical advisor[54] – to

[49] Shirey, 2000, p. 104 [184].

[50] Bellovin, 1989 [14]. Open BSD Security Advisory, 1997: `http://www.openbsd.org/advisories/res_random.txt`. Still a major issue in 2007: Klein, 2007 [110]; [111]. For MITM in the form of *Birthday attacks* on DNS header ID numbers cf. Stewart, 2007 [195].

[51] Example, 1997: `http://www.cert.org/advisories/CA-1997-22.html` (03-2008).

[52] *Drive-by Pharming*: `http://www.symantec.com/avcenter/reference/Driveby_Pharming.pdf` (03-2008)

[53] Honeynet Project, 2007 [90].

[54] Stajano, 2002 p. 51 [192].

prevent harmful drug mixes, this could involve more serious risks to personal safety. Similar risks would exist for business environments.

3.4.3 ONS Confidentiality

ONS, being based on DNS, provides no mechanism to achieve confidentiality. This lack applies to the query originator, the query target, and, most of all, the query content. As a part of an information architecture on physical objects, ONS involves even more risks to confidentiality and privacy than DNS does for surfing the Web. Before we discuss those problems in detail, two other DNS-related privacy problems must at least be mentioned: WHOIS, and geo-tracking via DNS.

Meta-information about domain owners has been and still is publically available,[55] which constitutes a debated privacy issue today.[56] This includes real names and addresses of the DNS or ONS server provider. For a manufacturer, this would probably pose no additional threat. However, in general IOT name service scenarios allowing for Discovery Services with arbitrary publishers, there might be situations where the information publisher prefers to keep his identity hidden (see Section 2.3.3).

On the other hand, the public nature of DNS information itself can be used for geo-tracking mobile hosts that have a domain name,[57] similar to tracking hosts via permanent IPv6 addresses. This problem will become more relevant in future UC environments where many devices would need a DNS name to enable remote service discovery and interaction.

For example, they could use future extensions of Multicast DNS (mDNS) and similar Rendevouz or *zeroconf* protocols,[58] which currently only use globally non-unique address space below the reserved `.local` TLD. The ONS object naming conventions could provide a simple and attractive global naming scheme for IP-enabled devices, which in turn could increase related ONS security risks, especially for confidentiality goals.

For the publisher, the ONS lack of confidentiality is evident: all the information published to ONS has to be considered public. There is no encryption or access control mechanism available. However, for the basic name service function that ONS is to provide, that is, the retrieval of manufacturer EPCIS addresses, this can hardly be considered a real risk, once the decision on using EPCs and to participate in the EPCglobal Network has been made. The linking of an object to its manufacturer will in general pose no confidentiality problem beyond an already established public

[55] Originally via the WHOIS protocol, RFC 3912, but also via Web-based search engines, for example: `http://www.domaintools.com/` (03.2008).

[56] ICANN: `http://gnso.icann.org/drafts/icann-whois-wg-report-final-1-9.pdf` (03.2008).

[57] Guha and Francis, 2007 [79].

[58] Multicast DNS: `http://www.multicastdns.org/` (03.2008).

EPC Manager number, and the actual object information stored at EPCIS can be protected by access control. Only if further ONS delegation below the EPC Manager is implemented, for example to external servers for specific object classes, business information may leak.

In the following, the ONS confidentiality problems for the client are discussed in depth, because they can be considered critical for individual privacy in smart homes, as well as for corporate risks of information leakage.[59]

ONS Query Confidentiality

The DNS and ONS query content, as well as its source IP address, will pass the Internet in clear text. In many situations, however, the EPC of an RFID tag has to be regarded as highly sensitive information – be it in private,[60] or in business environments where product and raw material flows constitute valuable market information (see the client confidentiality requirements in Section 2.3.3).

Even if the complete serial number of the EPC is not known, the combination of object class and company identifier is enough to determine the kind of object to which it belongs. Captured EPCs can be used to identify assets of an entity, be it an individual, a household, a company or another organization. If someone happens to wear a rare item, or a rare combination of belongings, tracking him may be accomplished even without knowing the actual serial numbers, simply by using the object classes (cluster tracking).

Many different ideas for securing the wireless RFID tag to reader communication against unauthorized access and eavesdropping have been proposed.[61] However, most proposals to mitigate RFID privacy problems do not take into account what will happen to an EPC once it is determined by an authorized reading process. To use the information stored in the EPCglobal Network about a given EPC, one needs to locate the corresponding EPCIS servers first. Even if the connections to these servers are secured by using protocols like Transport Layer Security (TLS),[62] the initial ONS look-up process would have neither been authenticated nor encrypted.

The DNS encoded main part of the EPC, which identifies the asset categories, will first traverse every network between the middleware and a possibly local DNS server in clear text – this could include an insecure local wireless network. Depending on the configuration of ONS caching and resolution process, this partial EPC will also be transmitted to additional DNS servers in the resolution path, which could include DNS Root servers, DNS servers authoritative for `.com` and `onsepc.com`, the ONS

[59] First discussed in Fabian et al., 2005 [62].

[60] Garfinkel, 2002 [73]; Weis, 2003, [211]; Albrecht and McIntyre, 2005 [1]; Günther and Spiekermann, 2005 [81]; Garfinkel et al., 2005 [72]; Bauer et al., 2006 [12].

[61] For examples and surveys confer to Weis et al., 2004 [212]; Garfinkel et al., 2005 [72]; Juels, 2006 [98]; Rieback et al., 2006 [170]; Rotter, 2008 [175]; Avoine [9].

[62] RFC 4346, Dierks and Rescorla, 2006 [43].

Root, and further down the corresponding hierarchy,[63] until the resolving process finally gets to query a ONS server of the company that serves as main reference for the object in question – usually belonging to the manufacturer. All traversed Internet service providers and backbone carriers might capture the partial EPC – this also holds for network taps placed by governmental organizations of countries the packets may cross.

All of the ONS query logging and analysis can be achieved with tools and techniques already in use today, virtually without any risk and only very moderate effort on the collector's side. Some of those are:

- DNS server logs: For an example, the statement `logging` in the common BIND DNS server, used with the category `queries`, *"reports the client's IP address and port number, and the query name, class and type."*[64] The *query name* corresponds to the partial EPC.

- dsc: *dsc*[65] is a statistical tool specifically customized for very busy name servers, e.g., DNS Root or TLD servers. It is already in use for analyzing queries to several DNS Root servers today.

- Network analysis tools: Examples include *tcpdump*, *ethereal*, and *wireshark*, in common use by network administrators, programmers, and security analysts.

- Snort: Snort[66] is a very capable open source Intrusion Detection System (IDS), certainly able to efficiently detect and store DNS queries of interest from IP packets. A simple Snort rule for logging DNS traffic to the ONS Root – without possible further refinement of ONS content detection – would look like the following:

  ```
  alert udp $OBSERVED_NET any -> $ONS_ROOT 53 (msg:"ONS Query to
  ONS Root"; rev:1;)
  ```

- Hancock: Hancock[67] is a domain-specific language for analyzing massive transaction streams, for example mobile phone connections or HTTP requests. It allows the formulation of efficient signatures of user behavior, and has been successfully applied in analyzing hundreds of millions of transactions a day.

In recent years, the importance of *insider attacks* on companies has lead to an increased monitoring of IT system use in the working place. Another trend is the

[63] Liu and Albitz, 2006, pp. 27 [122].
[64] BIND 9 Man., Ch. 6: `http://www.isc.org/index.pl?/sw/bind/arm94/` (03.2008).
[65] dsc: `http://dns.measurement-factory.com/tools/dsc/` (03.2008).
[66] Snort: `http://snort.org/` (03.2008).
[67] Hancock: `http://www.research.att.com/~kfisher/hancock/` (03.2008). See also Cortes et al., 2004 [36] .

ongoing convergence of physical (e.g., cameras and door security systems) and logical (i.e., IT-) security systems, combining them using unified management and alerting consoles, data stores, and event detection. Both trends could increase the demand for *Physical Intrusion Detection Systems* (PHIDS) using RFID, for example to monitor employees, their habit, and their belongings, as well as general item flow into and out of corporate premises.[68]

In addition to building monitoring systems[69] using cameras, sensors, and RFID readers to monitor the passing of goods around a company for theft prevention, detection of weapons, drugs, policy violations, a backend monitoring tier would analyze ONS and EPCglobal Network traffic in smart corporate building and factories. This could be achieved by using classical IDS with new signatures. Trends like the outsourcing of security services to specialized providers could potentially create the nucleus for an inter-corporate surveillance infrastructure on RFID-equipped items, especially those carrying globally unique EPCs.

In the next section, we extend the discussion of confidentiality issues to a more general perspective, taking the whole EPCglobal Network architecture into account, before the chapter is closed with a comparison of the ONS architecture with the IOT name service requirements from Chapter 2.

3.4.4 Query Confidentiality in the EPCglobal Network

In this section, we generalize the discussion of query confidentiality to the whole EPCglobal Network, before returning to focus on ONS in the next Chapter. Query confidentiality is a critical requirement,[70] but lacking in official documents and some security assessments of the EPCglobal Network.[71]

During ONS resolution, all queried servers and Internet service providers on the path could capture and store the partial EPCs, as well as the origin, i.e., the source IP address, of the query. Currently, there are already pilot projects of ISP to analyze and profile customer's surfing behavior for marketing purposes,[72] which could easily be extended to EPCglobal Network traffic.

Discovery Service providers will be able to harvest the source IP and the *full* EPC from their log files. Even if the actual connection to an EPCIS server is encrypted, the EPCIS operator himself (e.g., the manufacturer) could compile profiles of the subset of EPCglobal Network users who query for information at this particular

[68] PHIDS using RFID was more extensively presented by the author at a GI IDS workshop in 2006, cf. Fabian, 2006 [56].

[69] Ivanov et al., 2007 [95].

[70] First stated for ONS in Fabian et al., 2005 [62], extended to the whole EPCglobal Network in Fabian and Günther, 2009 [58].

[71] Konidala et al., 2006 [113]; EPCglobal, 2007, pp. 52 [53].

[72] Cf. S. Northcutt: `http://www.sans.edu/resources/securitylab/superclick_privacy.php` (04.2008). See also the analysis of *Phorm* given by R. Clayton: `http://www.lightbluetouchpaper.org/2008/04/04/the-phorm-webwise-system/` (04.2008).

server. The initial DNS lookup for EPCIS name resolution could betray the object brand to an even larger set of adversaries.

The user coverage that a functional role – e.g., ONS Server, ONS Root, EPCIS Server – in the EPCglobal Network can achieve, varies. For a very general classification of potential adversaries in terms of user coverage see the corresponding Figure 3.4, where observed EPCs and tracking possibilities are depicted. A local

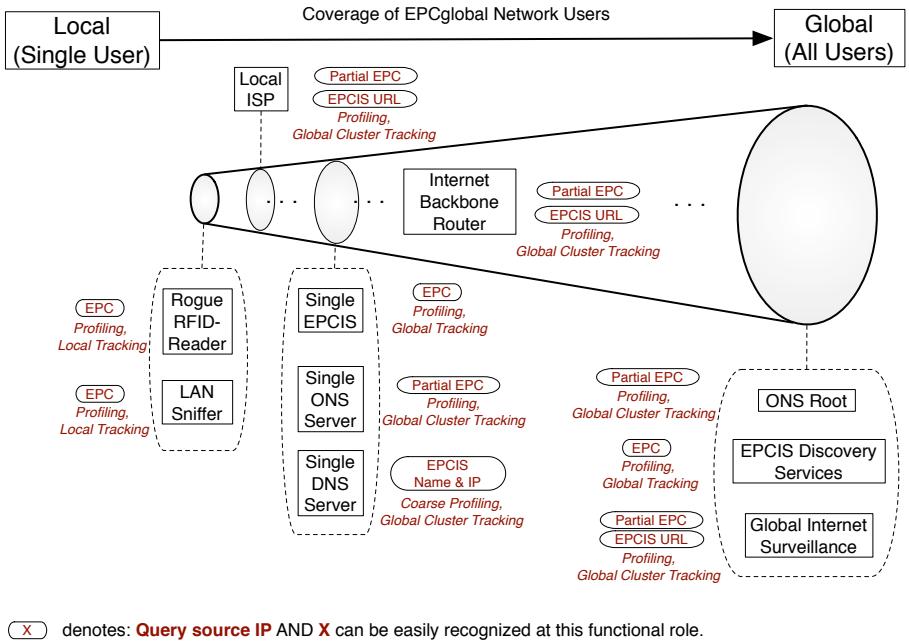


Figure 3.4: Adversary Coverage

ISP and other local adversaries will cover only a few users, the ONS Root virtually all – except for those who querying only for cached results by chance. In-between, there are Internet backbone routers, who will cover subset of users and subsets of targets, and single ONS servers, who see the subset of users interested in products of a specific manufacturer, as will EPCIS and DNS servers. It follows that attack trees,[73] which for example describe the profiling of someone's assets, will also have branches that represent several remote tactics (Fig. 3.5).[74]

If the EPCglobal Network becomes widely accepted, more and more business processes (B2B, B2C) as well as private applications will be able to use it without human intervention. This would leave those processes highly dependent on a robust and secure EPC resolution and information retrieval. In addition, it will expose them to potentially massive data collection by many possible couter-stakeholders and adversaries.

[73] Schneier, 1999 [180].
[74] In addition to local approaches identified in Spiekermann and Ziekow, 2005 [191].

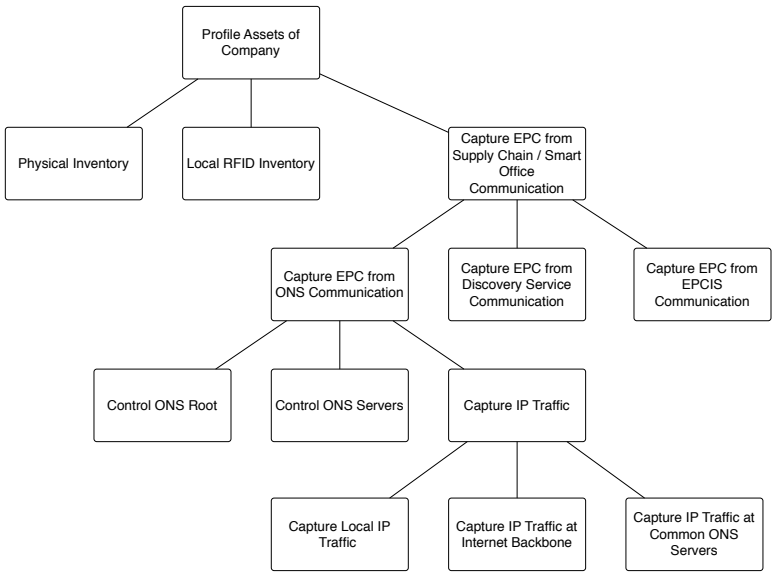


Figure 3.5: Example Attack Tree for Asset Profiling

3.5 Summary

Reflecting on the IOT name service requirements of Chapter 2, we can summarize the previous sections as follows. Of the functional requirements stated in Section 2.2, the *System Membership and Authorization* Procedure is conducted offline by EPCglobal. Corresponding online Subscriber Authentication services are planned by EPCglobal, but not yet published at the time of this writing. Access control for actual item information can be implemented by each EPC Manager itself at the EPCIS tier.

Publishing address information to ONS is only allowed for EPC Managers, that is, usually the manufacturers. Note that this is an organizational restriction, since technically ONS could transport NAPTR records pointing to several different parties. Correspondingly, *Querying* ONS will only be answered by pointers to manufacturer EPCIS. A lifting of this restriction is one of the tasks for EPCIS Discovery Services. ONS currently only works for *class-level*, partial SGTIN EPCs. Not providing a serial-level lookup is no necessary restriction, except for the anticipated load serial-level ONS could generate for the existing DNS infrastructure. Lookups for full EPCs of all types are planned for Discovery Services.

As a first approximation, it seems reasonable to assume that ONS inherits the main performance and scalability characteristics of DNS, which also can be considered as a lower bound for corresponding metrics because the EPCIS URL resolution step actually uses the standard DNS. In general, DNS seems to able to fulfill the scalability and performance requirements for class-level lookups, a more detailed discussion will be presented in Section 5.5.3. Robustness, however, suffers from limited redundancy and geographic dispersal of DNS data, especially for non-global

corporations that are not able to distribute their DNS servers. In addition, failure of servers high in the tree hierarchy could cripple large parts of the systems.

With respect to the security requirements of Section 2.3, ONS offers possibly the same level of availability as DNS at the level of leaf ONS servers, but potentially less at the level of the ONS Root compared to the DNS Root (see Section 4.2). ONS offers no integrity, and no confidentiality. In the next chapter, we study extensions and deployment strategies to enhance ONS security.

Chapter 4

Evolution: Enhancing ONS

> The people who can destroy a thing, they control it.
>
> *Paul Atreides*
> Dune, by Frank Herbert[a]

[a] Frank Herbert: Dune. Hodder and Stoughton, Paperback, 1993 (1968), p. 486.

4.1 Introduction

How secure and especially multipolar can ONS be made – without fundamental changes to its design? This question indicates the leitmotiv of the following chapter.

First we formulate and discuss the Multipolarity requirement for ONS. In addition, Multipolar ONS (MONS) is presented, a corresponding modification to the ONS architecture that guarantees multipolarity. This work, which has to the best of our knowledge no antecedents, was published in joint work in Evdokimov et al., 2008 [55]. Then, an analysis of possible security extensions and their applicability to ONS and EPCIS is presented, which is based on Fabian and Günther, 2009 [58]. Concerning related work for the latter, a survey of security measures for the EPCglobal Network was presented by Konidala et al., 2006, [113], but without considering client confidentiality requirements or multipolarity. Shih et al., 2005 [183], present a security framework for the EPCglobal Network based on Web service security standards, but focus on provider confidentiality requirements only.

The chapter is structured as follows. The first part discusses ONS unipolarity, and presents multipolarity extensions for ONS. The second part is dedicated to other countermeasures to further mitigate ONS security risks, and their applicability to ONS or EPCIS access.

4.2 Multipolar ONS

As was discussed in the previous Chapter 3, the Object Naming Service (ONS) is a central name service of the EPCglobal Network. Its main function is the address retrieval of manufacturer information services for a given Electronic Product Code (EPC) identifier. This allows dynamic and globally distributed information sharing for items equipped with EPC tags.

However, unlike in the DNS system, the ONS Root is *unipolar*; i.e., it could be controlled or blocked by a single country.[1] EPCglobal is delegating control of the root of the ONS hierarchy to a US-based company. Since RFID tags are foreseen by many to become ubiquitous and play a vital role in supply chains worldwide, such concentration of power in the hands of a single entity can lead to mistrust in the ONS, and may involve the introduction of proprietary services, increase in fixed costs, and loss of the benefits that an open, freely accessible, global system could bring.

A similar trend can be observed for global navigation satellite systems: In spite of the fact that the US-operated Global Positioning System (GPS) is globally available, free of charge, and even though deployment and maintenance costs are extremely high, various nations start or plan to introduce their own navigation systems to achieve more local control on an infrastructure deemed critical.[2]

To prevent a similar fragmentation scenario for the ONS, it seems reasonable to modify the initial design to take the distribution of control between the participating parties into account, and make the ONS *multipolar* – in contrast to the existing unipolar design. In the following sections, we document the unipolar nature of ONS and propose several modifications to allow for multipolarity without fundamentally changing the existing design.[3] In addition, we discuss approaches that could make the proposed architecture more secure by ensuring integrity and authenticity of the data delivered.

4.2.1 Multipolarity

In the following, multipolarity in the IOTNS context is made precise, and a comparison of ONS and DNS with respect to multipolarity is presented.

[1] Transferring – and narrowing – this political term to the power structure within the IOT: *"Unipolarity in international politics describes a distribution of power in which there is one state with most of the cultural, economic, and military influence."* Wikipedia, s.v. *Polarity in International Relations*, [215] (05.2008).

[2] Example GPS alternatives are GLONASS and GALILEO.

[3] See our second line of research on P2P-ONS, Ch. 5; Fabian and Günther, 2007 [57].

ONS and Multipolarity

The ONS Root will formally be under control of the international consortium EPCglobal, but practically run by the US-based company VeriSign.[4] VeriSign is also known as a major certification authority for SSL/TLS, one of the DNS root operators, and maintainer of the very large `.com` domain.[5]

We abstract from these particular circumstances to a more general scenario. Let the ONS Root, as it is designed today, be controlled by a single company c belonging to a nation or group of closely allied nations A. At any given time and state of global politics, there exists the possibility for the government(s) of A to influence those actions of c that concern international relationships — this influence can be exerted either directly via laws, or indirectly via political or economic pressure.

Definition 1. ONS Blocking Attack. *The current design of the ONS would allow nation A – controlling the ONS Root – to conduct the following blocking attack against another nation B: The ONS Root could be configured to formally deny any information to clients originating in B, compliant to the ONS protocol, or simply ignore any query from IP addresses belonging to B. An even more efficient way would be to drop inbound ONS packets from B at border routers of A.*

The result of this attack would be stale information at all companies in B. Cached addresses of EPCIS could still be used, but cannot be easily updated anymore. To recover, B may consider building its own version of an ONS Root answering all local queries. However, to feed this new root information from alternative external sources would be tedious and probably very time-consuming.

There would be serious business drawbacks for companies in B during that time. Companies outside of B, for example in A, would only – and in the worst case for A – be affected if they heavily rely on business with B, due to probable retaliate blocking of EPCIS access from A by B, or stale data on B at the ONS Root – this corresponds to a virtual embargo situation. All other companies would not directly be affected, leading to a comparatively low risk for A.

In a highly connected global economy based on the EPCglobal Network this kind of attack, or even its threat, could be highly effective and more efficient than a general disruption of the global system. This should be prevented already by a design that spreads out the control of the ONS Root more evenly.

ONS queries and responses are transmitted in plaintext and can easily be read by an adversary who is able to intercept them.[6]

Definition 2. Traffic Eavesdropping and Analysis. *The control over the ONS Root allows A to eavesdrop on all ONS queries reaching the root name servers, and to*

[4] EPCglobal, 2005 [50]; VeriSign, 2005, p. 8 [203]; URL: `http://www.verisign.com/verisign-inc/news-and-events/news-archive/us-news-2004/page_000846.html` (05.2008).

[5] EPCglobal, 2005 [50].

[6] Fabian et al., 2005 [62]; Section 3.4.3.

gather global business intelligence about location and movements of items tagged with EPC tags virtually for free and without risk.[7]

Such attacks are relatively easy to launch, both technically and legally,[8] and could force parties concerned with their privacy to refuse ONS adoption and to look for alternative solutions.

DNS and Multipolarity

As was outlined in Section 3.3.2, the DNS consists of a hierarchy of DNS name servers, each responsible for resolving hostnames of Internet sites belonging to its zone or pointing to another DNS name server if delegation takes place. DNS name servers authoritative for TLDs (e.g. `.eu`, `.com`) are operated by domain name registries – organizations responsible for managing and technical operation of the TLDs.

The root name servers are operated by governmental agencies, commercial and non-profit organizations. The root zone is maintained by the US-based, non-profit Internet Corporation for Assigned Names and Numbers (ICANN). ICANN was contracted for this purpose by the US Department of Commerce, which thereby holds *de jure* control over the root namespace. Currently the root zone is served by only 13 logical root name servers, whose number cannot be increased easily due to technical limitations. However, many of those servers are in fact replicated across multiple geographical locations and are reachable via Anycast. Anycast is a routing scheme that allows to set up one-to-many correspondence between an IP address and several Internet sites so that when an actual communication takes place the optimal destination is chosen.[9] As a result, currently most of the physical root name servers are situated outside of the US, see Fig. 4.1 showing the situation at the end of 2007.[10]

The concentration of *de jure* control over the root namespace in hands of a single governmental entity is subject to constant criticism from the Internet community. In theory, this entity has the power to introduce any changes to the root zone file. However, due to the *de facto* dispersal and replication of the root zone, such changes must be propagated among all the other root name servers, many of which are beyond the authority of the entity controlling the root zone. In case the entity decides to abuse its power and introduces changes in the root zone by pursuing

[7] For partly humorous scenarios supporting this assessment in case of DNS, cf. K. Auerbach at CircleID, July 2007, *Google Buys VeriSign (not really)*: `http://www.circleid.com/posts/google_buys_verisign_not/`, and: `http://www.cavebear.com/cbblog-archives/000232.html` (04.2008).

[8] According to an amendment to Foreign Intelligence Surveillance Act (FISA), US intelligence is allowed to intercept electronic communication between US and non-US bodies if the communication passes across US-based networks (Protect America Act of 2007).

[9] For DNS use cf. RFC 3258.

[10] Gibbard, 2007 [76]. Fig. 4.1 was created initially by Patrik Faltstrom via Google Maps, see `http://stupid.domain.name/node/407` (03/2008). Most of these servers are listed at `http://www.root-servers.org/` (03/2008).

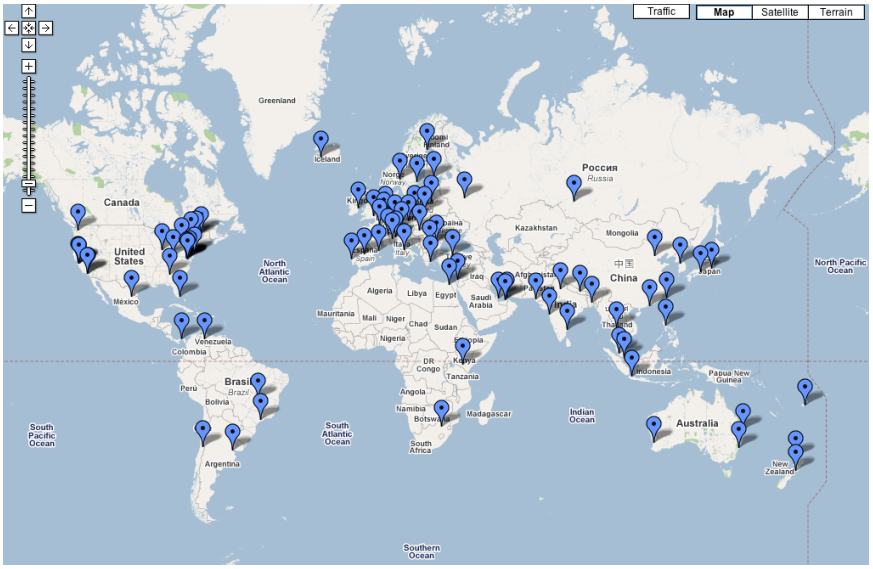


Figure 4.1: Geographical Distribution of DNS Root Servers

solely its own benefits, some of the root name servers may refuse to introduce the changes into their root zone files, which, in the end, may lead to the uncontrolled and permanent fragmentation of the Internet, undermining its basic principles and increasing business risk globally.

These consequences, as well as the fact that such changes have not occurred until now, allow to assume that the Internet is not directly dependent on the entity managing the root namespace, and that it is unlikely for this entity to introduce any changes impeding fair and global Internet access. As a consequence, unlike with ONS, the Blocking Attack is not realistic with DNS without severe risks to the initiating country.

4.2.2 Multipolar ONS Architecture

In this section we propose modifications of the current ONS architecture that would allow to distribute the control over the ONS Root between several independent parties, thus, solving the issue of unilateral root control.

Replicated MONS

One of the main reasons why the DNS was chosen for implementing the EPC resolution is, probably, the alleviation of effort required to introduce the ONS on a global scale: The DNS is considered by many practitioners as a mature and time-proven

architecture.[11] Its choice allows to deploy the ONS using existing DNS software and rely on best practices accumulated during decades of the DNS being in use. As a result, the deployment of a local ONS name server can be relatively easily performed by a system administrator with DNS experience using freely available software. Thus, if we want to modify the existing ONS architecture, it makes sense to initially try to stay consistent with the DNS protocol.[12]

Figure 4.2: VeriSign and ONS Root (Conceptual Picture)

The ONS Root will run on six locally distributed server constellations, all operated by VeriSign, cf. Fig. 4.2, a conceptual picture not showing the actual locations.[13] This strongly contrasts with the DNS architecture, where the root name servers are operated also by numerous other entities.[14]

A straightforward approach to avoid the unipolarity of the ONS is to replicate the ONS Root between a number of servers operated by independent entities, and to synchronize the instances of the root zone file with a master copy published by EPCglobal. To restrict the amounts of incoming queries, each root name server could be configured to cover a certain area in the IP topology and respond only to queries originating from there.

Such replicated ONS Root name servers could provide their services in parallel with the global ONS Root operated by VeriSign. The resolving ONS servers of organizations and Internet Service Providers (ISP) should be configured on the one hand with the domain name or IP address of the global ONS Root (`onsepc.com`), or, more efficiently, of the server responsible for SGTIN (`sgtin.id.onsepc.com`), on the other hand also with the corresponding replicated ONS server (e.g. `sgtin.id.onsepc-replication.eu`), potentially avoiding Anycast constructions like those used as later add-ons for DNS.

To evaluate the feasibility of this approach and the amount of data that has to be

[11] For dissenting arguments, however, see e.g. Ramasubramanian and Sirer, 2004 [161]; also cf. Section 5.5.3. For the DNS lack of security cf. Section 3.4.

[12] In the next sections we will temporarily ignore the severe ONS confidentiality issues already identified in Ch. 3.

[13] According to public information available at the time of this writing, some future changes in the ONS root server distribution are possible.

[14] Gibbard, 2007 [76].

replicated, we approximately calculate the size of the ONS Root zone file by estimating the number of RRs stored there, which define mappings between Company Prefixes and domain names of the corresponding ONS name servers. Today, there are about one million registered Company Prefixes.[15] We assume that at a certain time in the future most of them will have corresponding EPCIS services. The ONS Root zone file is a plain text file consisting of a number of `NS` RRs. As an example, consider an EPC number `400453.1734.108265` that can be resolved into one of two ONS name servers:

```
1737.400453.sgtin.onsepc.com IN NS ons1.company.com
1737.400453.sgtin.onsepc.com IN NS ons2.company.com
```

`IN` stands for Internet, and `NS` indicates that the record defines a name server authoritative for the domain. The number of name servers responsible for the same zone cannot exceed thirteen, and the DNS specification recommends having at least two. In practice, however, their number usually varies from two to five.

Assuming the average number of ONS name servers per company (c) as four, the average length of an NS record (l) as 60 symbols, and that one symbol takes one byte, and the number of registered Company Prefixes (p) as one million, we can roughly estimate the size R of the ONS Root zone file containing the RRs for all currently registered EAN.UCC Company Prefixes as

$$R = c \cdot l \cdot p, \tag{4.1}$$

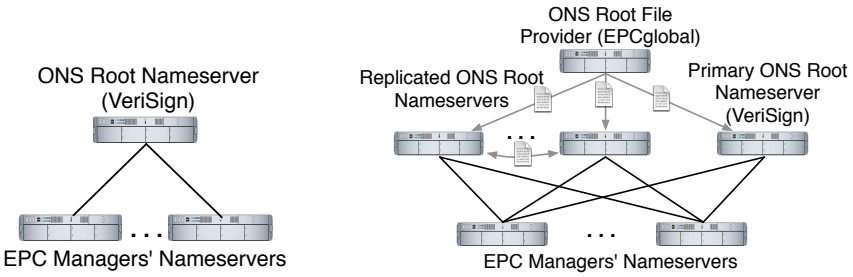
which is slightly above 200 megabytes. By using compression a text file may be reduced to 10-20% of its original size.

Thus we conclude that the distribution and regular renewal of the root file presents no technical difficulties. The master root file can be shared between ONS Roots by the means a simple file transfer or by a controlled instance of a peer-to-peer file sharing protocol like BitTorrent[16] that is frequently – and legally – used for large data files like scientific data or Linux distributions. The architecture is illustrated at Fig. 4.3(b) and will be further referred to as Replicated MONS.

The key requirement of Replicated MONS is the public availability of the ONS Root file. As soon as the root file is published and regularly updated, the replicated roots can be deployed independently from each other. In case those new roots will be configured to cover only certain areas, locations beyond their bounds will still be able to use VeriSign's name servers, remaining vulnerable to the Blocking Attack.

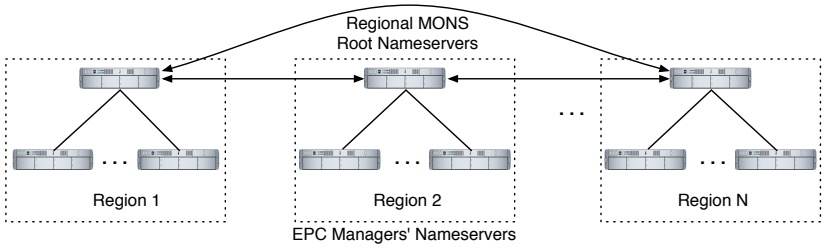
[15] According to GS1: `http://www.gs1.org/productssolutions/barcodes/implementation/` (09/2007).

[16] BitTorrent: `http://www.bittorrent.com/` (03/2008).



(a) Current ONS

(b) Replicated MONS



(c) Regional MONS

Figure 4.3: MONS Architectures

Regional MONS

The architecture described in the previous section provides a solution which allows any entity to maintain a copy of an ONS Root name server, enhancing the availability of the ONS. However, due to the necessity to cope with a high load, such name servers might not be accessible globally, potentially resulting in a – from a global perspective – unstructured patchwork of areas with ONS Root redundancy.

The high load on the root name servers will be mainly caused by the size and frequent updates of the root zone file. Compared to the DNS root zone file, which contains RRs on about 1500 TLD name servers and currently has a size of about 72 kilobytes at the time of this writing,[17] the ONS Root zone file will contain RRs for *all* EPC Managers' ONS name servers registered at EPCglobal. With RFID becoming ubiquitous, their number is expected to grow rapidly, resulting in millions of RRs. Also, due to a higher volatility of ONS Root RRs, their TTL parameters might be assigned lower values as compared to the RRs of the DNS root. As a result, the ONS RRs will be cached for shorter periods of time and a larger number of queries will be reaching the ONS Root name servers.

In this section we suggest a more radical alteration of the existing ONS architecture that will allow to reduce the size of the root zone file and the frequency of its updates by splitting it between a number of *regional root name servers*, at the

[17]The file is available from InterNIC: `http://www.internic.net/zones/` (03.2008).

same time offering a structured way to achieve area coverage for resolution. In this solution, a zone file of each regional name server contains RRs that correspond to EPC Managers belonging to a region for which a name server is authoritative. The membership to a region might be determined by a company's registration address, regional GS1 department that issued the Company Prefix, or other properties.

The architecture is depicted in Fig. 4.3(c), while the resolution process is presented in Fig. 4.4. In case the resolving name server and the EPC Manager – who corresponds to the EPC being resolved – belong to the same region ($n = m$), the step 2 is omitted and the resolution process is almost identical to the one depicted in Fig. 3.3: The regional root name server delegates the query to the name server of the EPC Manager, which returns the address of the EPCIS. However, if $n \neq m$, the query is redirected to the regional root name server authoritative for the Region n (step 2), which in turn delegates it to the name server of the EPC Manager. We will refer to this architecture as *Regional MONS*.

Compared to the ONS resolution process described in Section 3.3.4, the case of delegating a query from one regional ONS name server to another (step 2) introduces an additional resolution step. Consequently, this requires an extension of the EPC scheme and the introduction of a new prefix that will be resolved at this step.

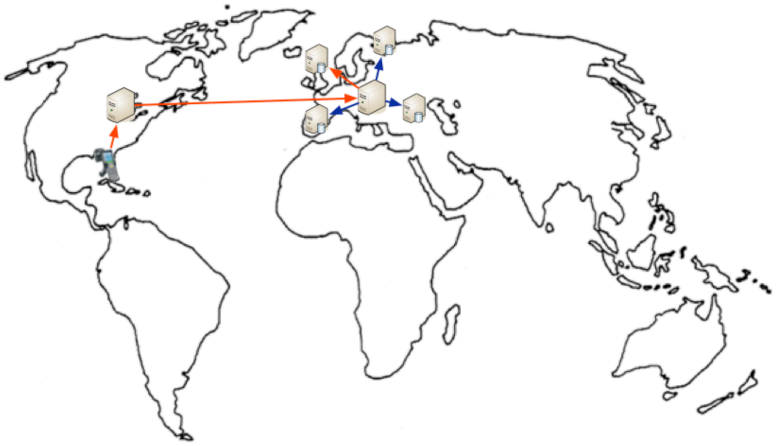
Following the approach for constructing an EPC, a natural choice would be a *regional prefix* pointing to a country or a region of origin for a given product. The introduction of this regional prefix requires an update of the EPC encoding standards, which might result in a lengthy and costly process. However, the EPC encoding schemes already contain enough information to unambiguously associate an EPC with a certain region.[18] The first three digits of the EAN.UCC Company Prefix identify the country of GS1 membership for the company, for example 060–099 for the US and Canada (0718908: Apple Inc.), 400–440 for Germany (4009700: Danone GmbH). Therefore, an alternative to the introduction of a new regional prefix field would be to use these digits for associating EPC identifiers with corresponding regions. Each regional root name server will be responsible for one or several regional prefixes.

Note that a resolver still sees the Regional MONS architecture as a hierarchy: the MONS Root of its region is being perceived as the root of the whole hierarchy (Fig. 4.6). We call such a structure a *relative hierarchy*. A regional name server authoritative for a region from which the resolution takes place is called its *relative root*. This allows for implementing Regional MONS within the DNS framework, reflecting the approach described in the ONS specification.

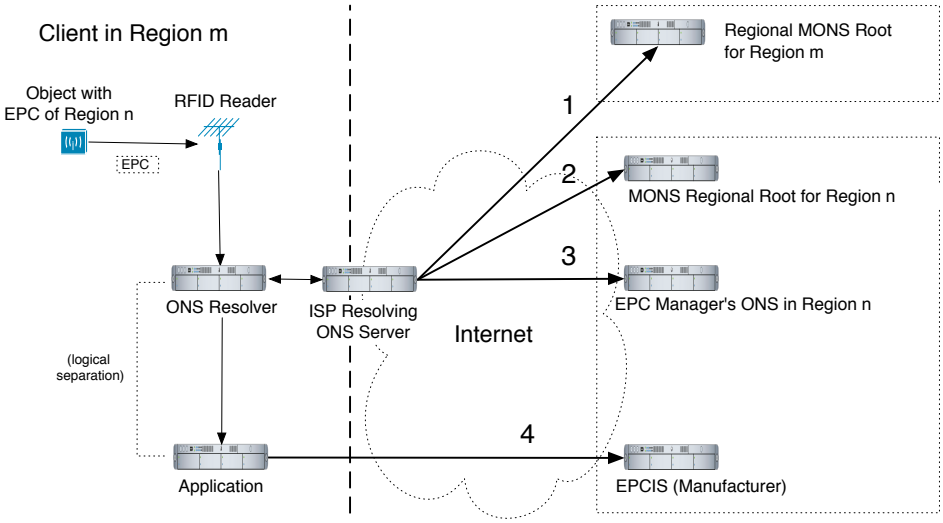
In the following, we assume that the regional prefix is defined as the first three digits of the Company Prefix. To access an EPCIS that could provide data about a given EPC identifier, the identifier is like with ONS translated into a DNS-compatible address, but now the first three digits of the Company Prefix have to be explicitly

[18] EPC encoding schemes are defined in EPCglobal, 2007 [51].

(a) Regional MONS Root Scope



(b) International Query



(c) Regional MONS Resolution Process

Figure 4.4: Regional MONS

Header	Filter Value	Partition	Regional Prefix	Company Prefix (EPC Manager)	Item Reference (Object Class)	Serial Number
8 Bits	3 Bits	3 Bits	8 Bits	20-40 Bits	4-24 Bits	38 Bits

Figure 4.5: EPC Regional Prefix

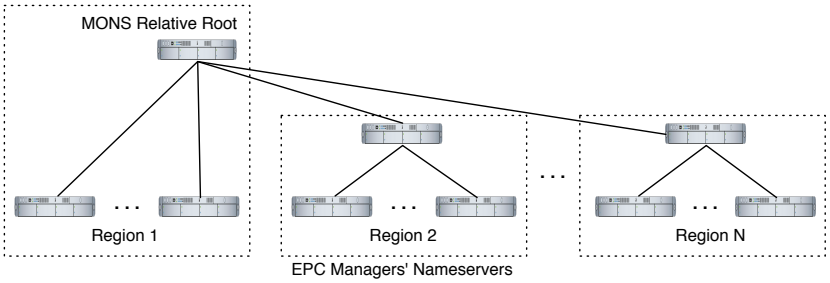


Figure 4.6: Relative Hierarchy of Regional MONS Name Servers

separated by dots and placed to the right of the rest of the inverted EPC (e.g. `1734.453.400.sgtin.id.onsepc.com`). Assume that the domain name of the regional name server authoritative for zone `400.sgtin.id.onsepc.com` is `ns1.mons.eu`. An ONS client physically located at the same region is configured to sends all its ONS queries to `ns1.mons.eu` (step 1 at Fig. 4.4), which it views as the relative root of the Regional MONS.

Correspondingly, a resolver that belongs to a different region will be configured with the address of a different regional root, also viewed as a relative root. In this example, we deliberately choose the domain name of the regional root to have the TLD (`.eu`) corresponding to the region of its authority. This avoids the dependency on foreign entities administering the domain names of the regional name servers and excludes the possibility of a Blocking Attack from their side.

Note that the resolution process described above does not require an EPC to be translated to the domain name resolvable by the DNS of the Internet. The only domains relevant to the ONS resolution are the dot-separated EPC and the domain pointing out in which format an EPC number is stored. This makes the three rightmost domains abundant, since `1734.453.400.sgtin` would be already sufficient for unambiguous ONS resolution.

By appointing specific name servers to regions, Regional MONS naturally shifts the load to name servers authoritative for economically developed or industrial countries, since regional prefixes of such regions will occur on the majority of the EPC identifiers. Moreover, regions whose export values are too low, or who are not interested in maintaining their own Regional MONS Root name servers could delegate this responsibility to third parties, as it is sometimes done with country code

TLDs.[19] Once their situation changes, they can take back their reserved share of the system by a minor change in the table of Regional MONS Roots (MONS Root Zone).

4.2.3 MONS Prototype

In this section we present a possible fragment of the Regional MONS architecture implemented using BIND DNS Server software. BIND (Berkeley Internet Name Domain) is the most common DNS server in the Internet and the *de facto* standard for Unix-based systems. ONS can be deployed using standard DNS software, so it is very likely that a considerable portion of ONS name servers will be using BIND.

In our sample scenario we consider two regions with regional codes 400 and 450 and two EPCISs, each providing information about one of the following SGTIN formatted EPC identifiers: 400453.1734.108 and 450321.1235.304.

The main configuration file of a BIND server is the `named.conf`. RRs for namespaces are stored in zone files often named *`namespace`*`.db`. Fig. 4.7 presents a possible configuration of four ONS name servers that constitute this fragment of the Regional MONS hierarchy. The fragment includes two regional MONS Root name servers authoritative for regional prefixes 400 and 450, correspondingly, and two name servers of EPC Managers.[20]

The regional roots are configured as relative roots of the `sgtin` zone and as authorities for the respective regional codes (`400.sgtin` and `450.sgtin`, correspondingly). The `sgtin.db` file describes the relative root zone (`sgtin`) by declaring the name server as the authority for this zone and referring to the content of `onsroots.db` file, which represents the MONS Root Zone. This file is the same for all regional roots and defines the delegation of the zones (using the regional codes) to the regional roots. The RRs of the `400.sgtin.db` and `450.sgtin.db` files introduce a further delegation step by pointing to the name servers of the respective EPC Managers that complete the resolution process by returning the URI of the requested EPCIS via NAPTR RRs.

To make the zone files less dependent on infrastructure changes in the MONS hierarchy, they may contain only NS records without mentioning the corresponding IP addresses in A records. Therefore, if one or several name servers has its IP address changed the zone files still remain consistent. However, this can prolong the resolution process, since ONS name servers will have to query the DNS to resolve domain names to IP addresses.

[19] Gibbard, 2007 [76].

[20] Note that all domain names, IP addresses, and URIs in this example are fictional.

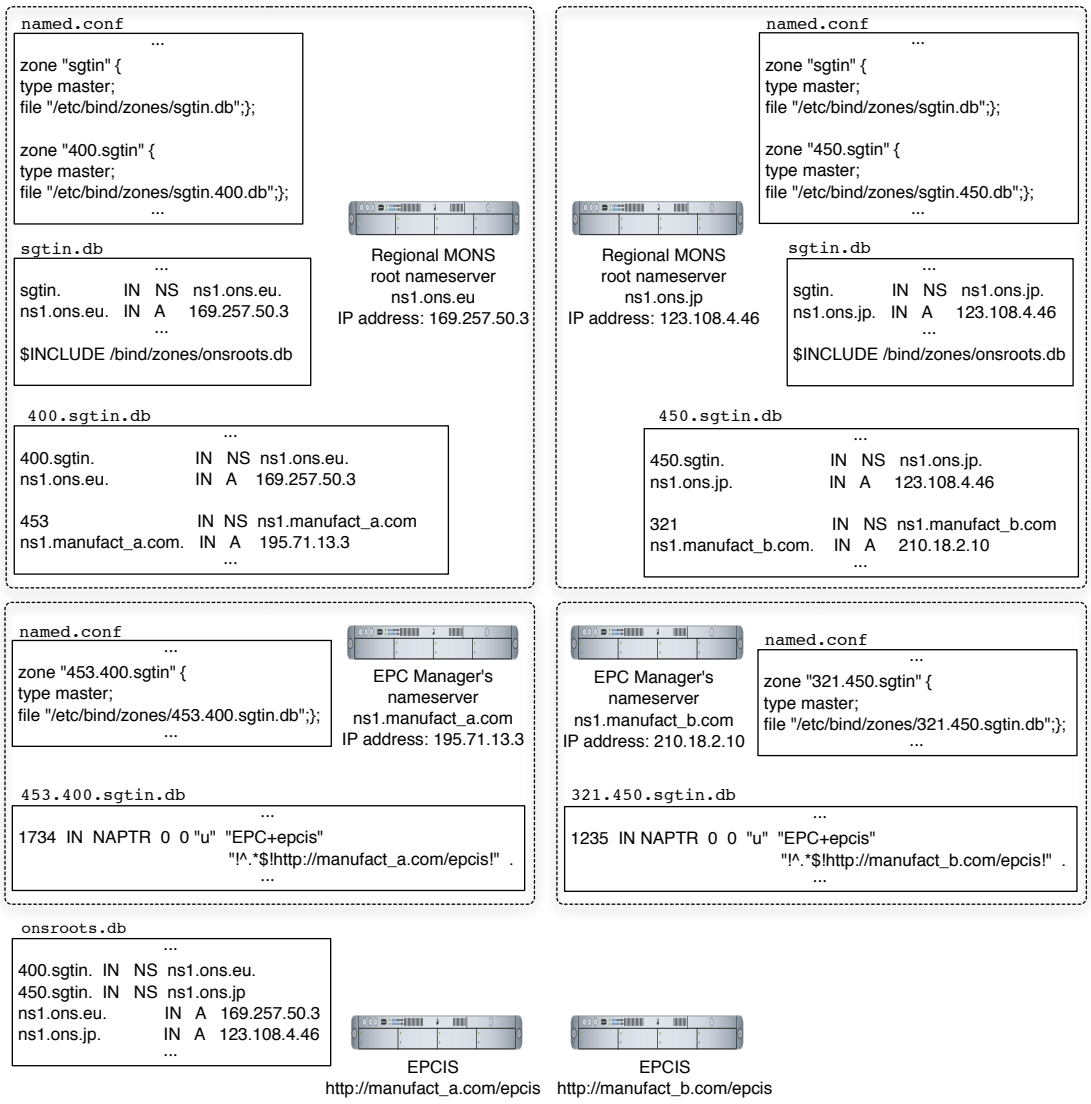


Figure 4.7: Example Regional MONS Hierarchy

4.2.4 Modularity

One further advantage of Regional MONS is that each region could implement different resolution architectures for its own subsystem below the root zone. For example (see Fig. 4.8), a region r could use the original ONS specification based on the DNS, another region n could use a centralized IOTNS, while yet other regions, like m, could implement subsystems based on Distributed Hash Tables (DHT), e.g. the OIDA system presented in Chapter 5.[21]

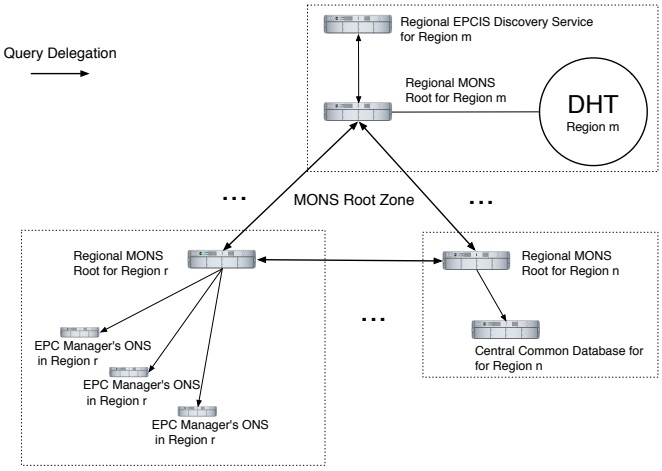


Figure 4.8: Modularity of MONS Subsystems

Delegation between MONS and heterogeneous subsystems can be established by bridging nodes that are able to use both protocols. In the DHT case for example, a DHT node queried by external DNS clients uses the DNS protocol to answer. However, to communicate with other DHT nodes, the specific overlay network communication is used, for example as defined in the Chord DHT.[22] This combination of DNS and DHT has been successfully implemented for general DNS use, for example in CoDoNS.[23]

4.2.5 Conclusion

In this section we presented MONS, a practical architecture to achieve multipolarity in the ONS. We also showed how multipolarity in corresponding authentication extensions can be achieved. To our knowledge, this is the first extensive discussion and solution proposal of the multipolarity problem for ONS, which in a future Internet of Things may have even more detrimental consequences than the analogous problem currently debated for DNS.[24]

[21] Fabian and Günther, 2007 [57].
[22] Stoica et al., 2003 [197].
[23] Ramasubramanian and Sirer, 2004 [161].
[24] Kuerbis and Mueller, 2007 [114].

On the policy side, analysis of the practical political and administrative challenges of distributing control over the ONS is an important line for future research. Not last, there is urgent need to solve further multilateral security problems of ONS and related systems like MONS, especially their possible impact on corporate and individual privacy.

We turn now to ONS and MONS integrity risks, and will discuss strategies for their mitigation.

4.3 Protecting Integrity: ONSSEC

Today's Internet must be regarded as a highly insecure environment, a fact that has been acknowledged not only by the security community, but also political institutions.[25]

Surprisingly, security measures have not been considered intrinsically from the beginning in the EPCglobal architecture standards,[26] but seem to be held as optional and mostly to be added later by its users.[27] Besides availability and confidentiality risks of the EPCglobal Network and the ONS in particular, a major concern is the lack of authentication methods in the current ONS standard.

Without additional security measures, global business systems depending on the ONS, as it has been designed in the standard so far, could suffer from cache poisoning and MITM attacks,[28] leading to spoofed EPCIS address information, and potentially also to forged EPC information, or via additional vulnerabilities, malware infection initiated by malicious servers. Adding countermeasures like DNS Security Extensions (DNSSEC) later, however, will also have an impact on properties of the whole system, like performance, security and privacy, as well as multipolarity.

In this section we first take a short look at the recent DNSSEC standards, discuss how DNSSEC could be used to secure ONS data, resulting in a substructure of DNSEC we propose to call ONSSEC. Finally we suggest mechanisms to achieve multipolarity for ONSSEC.

4.3.1 DNSSEC

To address the lack of authentication in the DNS, a set of mechanisms called DNSSEC (DNS Security Extensions) has been designed.[29] DNSSEC provides data

[25] For a notable example from the USA cf. to this President's Information Technology Advisory Committee (PITAC) report from 2005: PITAC, 2005 [158].

[26] Fabian et al., 2005 [62].

[27] EPCglobal, 2007 [53].

[28] Atkins and Austein, 2004 [7]; see Section 3.4.2.

[29] The recent version of DNSSEC is presented in RFC 4033, Arends et al., 2005 [5], and related other RFCs.

integrity and authenticity for the delivered DNS information by using public-key cryptography to sign sets of resource records (RRs). It uses four resource record types: Resource Record Signature (RRSIG), DNS Public Key (DNSKEY), Delegation Signer (DS), and Next Secure (NSEC), the last one is used to provide authenticated denial of existence of a zone entry.[30] Each DNS zone maintainer is also responsible for providing a signature of those zone files. These signatures are stored in an RRSIG record. The server's public key could be transferred out-of-band, or be stored and delivered via DNS itself using an RR of type DNSKEY.

The use of separate zone-signing and key-signing keys enables easy resigning of zone data without involving an administrator of the parent zone.[31] However, having a signature and an apparently corresponding public key does not guarantee authenticity of the data – the public key and identity must be securely linked by a trusted entity, most practically, by the maintainer of the respective parent zone. To be able to verify an arbitrary DNS public key in a scalable way, chains of trust down from the – necessarily trusted – root of the DNS would be necessary, where each parent DNS server signs the keys of its children, after having verified its correspondence to the correct identity by some external means.

Even after a major redesign in 2005,[32] DNSSEC is not yet widely established throughout the Internet, though recent developments like the signing of some countries' TLD seem to indicate better chances of its adoption.[33] Reasons for the slow DNSSEC diffusion include, first of all, reluctance to major changes for critical services like DNS, scalability problems of key management, increased message size, computational and memory overhead, and the administrative problem of building chains of trust between servers of many different organizations. None of those problems, however, seem completely intractable in the future. But similar to most diffusion processes depending on network effects, there is also the problem of establishing a critical mass of DNSSEC users with different incentives.[34]

With respect to confidentiality, however, even if DNSSEC could be widely configured to actually encrypt the DNS information, which is not a stated goal so far,[35] the company prefix of a given EPC could still be guessed by following the sequence of IP addresses the ONS queries are sent to. No measures for increasing the availability of ONS servers are offered by DNSSEC, on the contrary – signature checking introduces additional load to the involved servers.[36] Despite these problems, the establishment of a new global business architecture like the EPCglobal Network could be a major opportunity to launch ONSSEC, the adaption and restriction of

[30] For details cf. Arends et al., 2005 [5].
[31] Liu and Albitz, 2006, pp. 335 [122].
[32] With RFC 4033, Arends et al., 2005 [5], which replaces RFC 2535 from 1999 that in turn rendered the original RFC 2065 from 1997 obsolete.
[33] Friedlander et al., 2007, [70].
[34] Ozment and Schechter, 2006 [146].
[35] Arends et al., 2005, Section 4, p. 8 [5].
[36] Atkins and Austein, 2004 [7].

DNSSEC to ONS use. But DNSSEC suffers from a major unipolarity problem: Who should control the anchor of trust, the keys for the root zone? This problem must be solved for a multipolar ONS to avoid unwanted indirect unipolarity for MONS introduced by its security extensions.

4.3.2 ONSSEC

DNSSEC can be applied to MONS as follows, cf. Fig. 4.9: Each Regional MONS Root provider signs the key-signing keys of all EPC Managers in its region. This is major administrative task and has to involve the verification of the EPC Manager's identity. This procedure is, however, less cumbersome than signing *all* subdomain keys of a given TLD, rendering ONSSEC introduction more scalable than general DNSSEC, where probably more delegation steps are also involved.

The EPC Managers are then able to sign their own zone-signing keys and the actual zone data. They can repeat the latter procedure after each change in zone data without contacting the Regional MONS Root; they are also able to periodically change their zone-signing keys for better long-term security. The EPC Manager's name servers can now answer MONS queries by returning the actual zone information in combination with the signature. This signature can be verified by a client by retrieving the public key of the regional MONS Root.

Here another (cf. Section 4.2.3), bigger problem of using the flexible option of general DNS names in (M)ONS resource records becomes apparent (e.g. in URIs of NAPTR records for EPCIS, see Fig. 4.7): Without an established global trust structure and ubiquitous use of DNSSEC, arbitrary DNS names and resolution steps would not easily be covered by authentication measures. As long as this situation holds, the tradeoff between flexibility vs. lack of authenticity needs to be constantly evaluated.

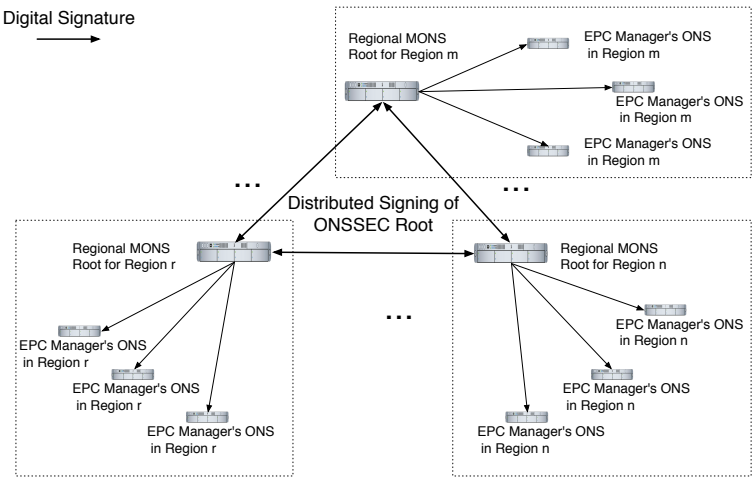


Figure 4.9: Multipolar ONSSEC Trust Structure

With the described Regional MONS architecture, there would be multiple roots of trust. This situation could be impractical because clients who often resolve EPCs of foreign regions would have to trust multiple public keys, those of the local and all foreign regional MONS Roots. With DNSSEC, it is often stated as best practice for a single entity to control the root zone key signing keys.

It is, however, a subject of current international debate which organization should represent this entity – for example, interest has been expressed by US authorities like the Department of Homeland Security.[37] A similar problem exists for the MONS Root zone.[38] In the following section, we briefly discuss options for a solution.

4.3.3 Multipolar ONSSEC

Multipolarity for the root key control of ONSSEC – that is DNS Security Extensions applied to (M)ONS – could be achieved by multiple signatures, that is, each regional MONS Root would sign the root zone,[39] or more elegantly and possibly with better scalability, by the use of one virtual ONSSEC root by applying threshold cryptography.

An *(n, t)-threshold cryptography scheme* allows n parties to share the ability to perform a cryptographic operation (e.g., applying a digital signature), so that t ($t \leq n$) parties can perform this operation jointly, but at most $t - 1$ (malicious) parties are not able to do so, even by collusion.[40] Famous threshold secret sharing schemes include Shamir, 1979 [182], using polynomial interpolation, and Blakley, 1979 [16], based on intersection of n-dimensional hyperplanes. Secret sharing could be used to share the private key of the virtual ONSSEC root, but once used, the entire private key may be compromised.

More secure are threshold function sharing schemes, extensions of the basic secret sharing, which allow for digital signatures without letting a single party know the complete key during operations.[41] The signing of the regional root keys and the MONS Root zone should be quite a rare operation in comparison to the signing of actual manufacturer zone data. Therefore, these schemes could be implemented without major performance penalties on the whole system.

In summary, using threshold cryptography would enable the distributed and multipolar signing of the MONS regional root keys (Fig. 4.9), as well as the MONS Root zone that contains address data of all Regional MONS Roots.

---

[37] Leyden, 2007 [120].

[38] The `onsroots.db` of the prototype in Section 4.2.3.

[39] Kuerbis and Mueller, 2007 [114].

[40] Menezes et al., 1997, pp. 525 [131].

[41] See e.g. Shoup, 2000 [185]; Kaya and Selcuk, 2007 [108] for schemes with usable performance properties.

4.4 Further ONS Risk Mitigation

Even though the *unipolarity problem* can be solved by MONS, there still will be only limited redundancy for regional roots and single ONS servers to fulfill *availability* requirements. And even though *integrity* requirements may be satisfied using ONSSEC, there are still open security problems regarding *confidentiality*.

Still all (M)ONS communication happens in clear text. All ISPs and (M)ONS servers can read the queries and responses, see Fig. 4.10. Instead of one single, global "Big Brother" with ONS – i.e., the ONS Root – there will be multiple regional Big Brothers with MONS. If confined to ONS queries, not a single one of them will see all (IP, EPC)-tuples that are issued to the ONS, but will cover a fraction proportional to the regional population of ONS clients, as well as all queries from foreign MONS Root for objects registered in its region.

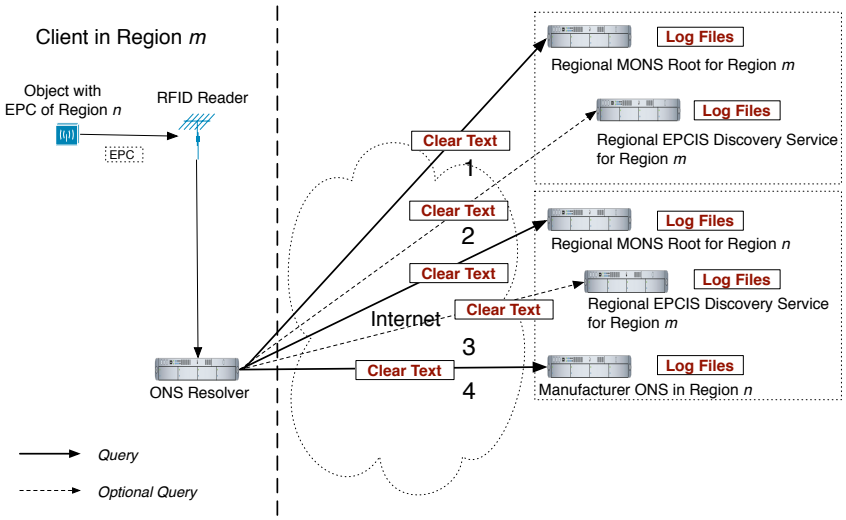


Figure 4.10: MONS Query Confidentiality Issues

In the following sections, we discuss possible countermeasures to reduce the confidentiality problem.[42] All of the following is formulated for ONS, but is also applicable to MONS.

4.4.1 Network Design

Larger enterprises may be able to reduce risks to IOTNS query confidentiality by using a well-designed internal network structure, especially in case of ONS a carefully planned DNS server hierarchy. Split DNS[43] may be deployed to separate internal from externally viewable name spaces, so that at least company-internal queries will

[42] The following section extends Fabian et al., 2005 [62]; Fabian and Günther, 2009 [58].
[43] BIND Manual: `http://www.isc.org/sw/bind/arm94/Bv9ARM.ch04.html` (03.2008).

remain confidential, and external parties cannot investigate internal name spaces. However, this is only helpful if the company uses a large percentage of items for which it is the EPC Manager; e.g., that are produced by it.

In other cases, centralization strategies for ONS queries could be helpful. All ONS queries from internal machines at any company site could be forwarded – preferably using Virtual Private Networks (VPN) – to a central company DNS server, which in turn performs the external resolution process. Even then all the EPCs that are resolved by the company could be intercepted outside of the Intranet borders, but not easily assigned to particular locations – though an attacker might apply a careful analysis of time, possibly combining this information with captured EPCs from region-specific objects.

For an attack in a realistic application scenario, consider a company using smart offices with ubiquitous RFID readers where outsiders might witness the introduction and the actual kind of new items – such as newly introduced laptops of a specific manufacturer – anywhere in the enterprise.

If a company just uses an internal and private version of the EPCglobal Network without depending on outside information – for example, if only self-manufactured items are of interest – no EPC leakage to outsiders would occur and risks to integrity and availability could be limited likewise to internal attackers. But this special case would deprive the company of the intended advantages of a global and dynamically updated EPCglobal Network, as only company-internal data sources about EPCs could be accessed.

Another countermeasure could be the prolonging of ONS and EPCIS caching times to reduce the frequency of the EPC crossing the Internet. Depending on the application scenario, the EPCIS dynamics, and the demand for fresh information, risk-reducing caching strategies may be viable.

4.4.2 VPN and TLS

The idea of concentrating ONS queries to prevent an exact locating of the corresponding items could be extended to small networks of trusted business partners (or neighbors in smart homes) by forming a so-called *Extranet*[44] (Fig. 4.11), connected by Virtual Private Networks (VPN) using IP Security[45] or Secure Sockets Layer (SSL), in newer versions called Transport Layer Security.[46]

All parties could connect to a central ONS resolving server via VPN, and this server issues the ONS queries to the outside world. Beyond this point, no protection by VPN would be practical, if access to many different third parties beyond the borders of the extranet is required, because the possible communication partners are nearly

[44] Cheswick et al., 2003, p. 247 [29].
[45] IPsec, RFC 4301, uses encryption on the Network Layer.
[46] TLS, RFC 4346, Dierks and Rescorla, 2006 [43].

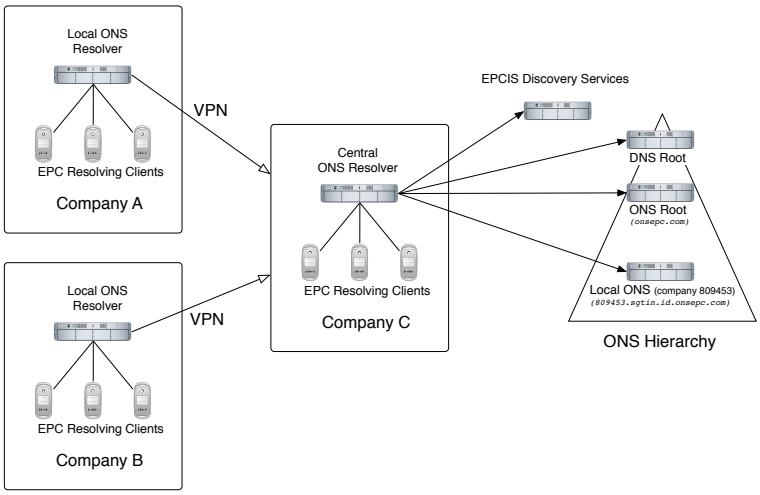


Figure 4.11: VPN and Extranets

countless and in general not known in advance – the problem of key management for building VPNs to every company that offers a relevant ONS and EPCIS server would render such solutions not scalable without an existing PKI, and even then subject to very high latency in case of ONS. Therefore, only closed groups of business partners who run their private version of an EPCglobal Network would be able to reduce their confidentiality and integrity risks significantly by using extranets. VPN scalability could change if there would be a PKI for the IOT available, which might be used for VPN key material and certificates.[47]

The deployment of an extranet could only limit threats to EPC confidentiality, but in the case of external information sources, not to integrity, or in general to ONS availability. In addition to issues of trust and administrative overhead, there will be an increased network load for the central party, depending on the scale of RFID reader deployment, caching strategies, and the intensity of usage of the EPCglobal Network by every single partner.

SSL or TLS could also be used without building a VPN, that is, without tunneling whole IP packets between networks, but for securing the application layer on a hop-by-hop basis only, like used for HTTP over SSL/TLS (RFC 2818) to secure web browsing. This mechanism originally only worked for TCP, but has recently been extended to UDP datagrams.[48] This would make it suitable for DNS and ONS use.

For each ONS delegation step, however, a new TLS connection would have to be established, which would negatively affect the performance of the ONS look-up process. At the current time, the CPU and memory overhead on the ONS Root

[47] The EPCglobal certificate standard indicates a future X.509-based PKI (RFCs 3280, 5280) for the EPCglobal Network, whose extent and scalability are yet to be determined, EPCglobal, 2008 [54].

[48] Datagram Transport Layer Security (DTLS), RFC 4347, Rescorla and Modadugu, 2006 [166].

caused by the necessarily vast amount of concurrent cryptographic operations does not seem to be practically feasible.

TLS would not help against adversaries who control ONS servers or the ONS Root, which would be communication endpoints, able to see the query and its orign. It could, however, definitely be deployed to reduce confidentiality and integrity problems of EPCIS communication against external adversaries (like an ISP), not the EPCIS provider, if an appropriate global trust structure between partners can be established.[49]

As with most security mechanisms, TLS has some usability problems, for example the validation of certificates,[50] malware-infected clients, lack of client-side certificate infrastructure, and resulting possible MITM attacks, which could be mitigated by protocol extensions.[51]

4.4.3 Mixes and Onion Routing

The culmination of the concentration strategy above, i.e., collecting ONS queries from different sources to hide the real source IP address, would be the use of so-called anonymous mixes,[52] a strategy that might be viable for supply chains as well as for private households in UC (Fig. 4.12).

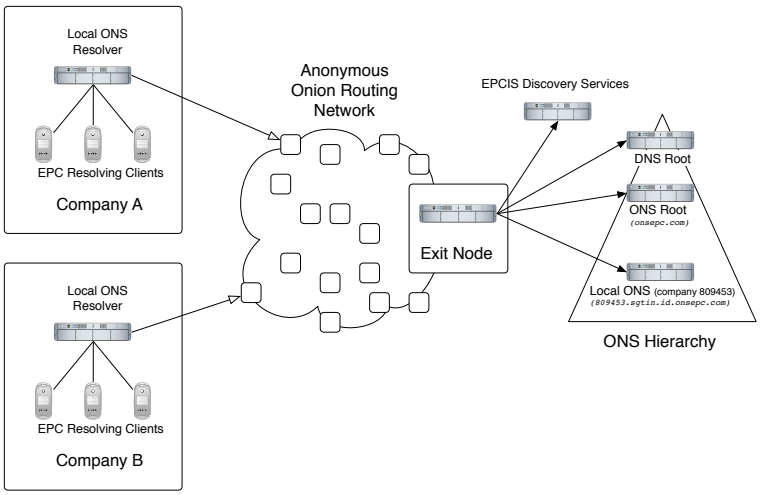


Figure 4.12: Onion Routing

The key idea of anonymous mixes, and the lower latency onion routing[53] – the

[49] As a side note, the leading player in the market for corresponding certificates is again VeriSign, cf. Netcraft SSL Survey, public data from 2006: `http://news.netcraft.com/SSL-survey` (04.2008).

[50] Ozment et al., 2006 [147].

[51] Oppliger et al., 2008 [145].

[52] Chaum, 1981 [28].

[53] Syverson et al., 1997 [198].

most popular implementation being Tor[54] – is to cryptographically transform and mix Internet traffic from many different sources, in order to hamper matching a particular IP packet to a particular source. Mixes usually store messages for a while before sending them out as batches, reducing the chance of input to output correlations. With onion routing, the transmitted data is wrapped into multiple encryption layers (like an onion) by using the public keys of the onion routers on the transmission path, but is not stored for later transmission, resulting in lower latency suitable for near real-time applications.

For ONS however, in some scenarios the usability of mixes, and perhaps also of the faster onion routing, could potentially be reduced by latency and performance issues, though more detailed performance studies should be conducted once, for example, Tor's handling of UDP has matured. Onion routing could also be used to anonymize traffic directed at EPCIS servers, and would be viable also for protecting private households. This could enhance anonymity and partially confidentiality, but not the integrity of the received messages. Tor could also be used for the whole EPCglobal Network traffic, but for EPCIS Discovery and EPCIS access conflicts between anonymity and identification needs for access control will need to be solved. For future use of an IOT in UC environments, Tor's *hidden server* functionality could be useful for censorship-resistent information by third parties about particular objects.

Besides anonymity, Tor also offers enhanced confidentiality as long as the traffic is inside the onion routing network, but not at the exit nodes and beyond.[55] However, at this point the source of the query is anonymized, so that the collection of (IP, EPC)-tuples for profiling is in general nearly impossible, except if the identity of the ONS client could be inferred from additional non-ONS traffic leaving the same exit node at the same time.

One additional caveat remains, however, that in some situations the observation of query time and EPC could be used for analysis based on adversarial background knowledge. For example, if an EPC is already related to an identity, its observation on the network or at ONS servers – including patterns over time – could indicate some activity or movement by its owner. To prevent this, some additional obfuscation of the EPC would be necessary.

Tor does not increase the integrity of received messages that originate outside of the onion routing network, nor could it increase an ONS server's availability, as any host offering services needs to be somehow addressable, and would therefore be potentially attackable by DoS. Known attacks on onion routing include timing correlations, traffic analysis,[56] intersection and predecessor attacks that are based

[54] Dingledine et al., 2004 [45].

[55] For attacks using data collection on clear text traffic at Tor exits nodes cf. Wired, Embassy E-mail Account Vulnerability Exposes Passport Data and Official Business Matters, August 31, 2007, URL: `http://blog.wired.com/27bstroke6/2007/08/embassy-e-mail-.html` (05.2008).

[56] Murdoch and Danezis, 2005 [136].

on observing the *churn*, that is joining and leaving of Tor nodes, and corresponding onion routing path reformations.[57] Many of these attacks, however, seem to apply to any practical anonymity system, and indicate boundaries for feasible anonymity in today's Internet environment.

Tor is a very important and perhaps the most mature approach for user privacy on the Internet. There are, however, ongoing arguments against its ability to protect – besides civil freedom – also criminal activities from law enforcement. We argue that a newly designed, privacy-critical service like ONS should offer some kind of client protection on its own, without depending on external and optional measures.

4.4.4 Private Information Retrieval

Methods from Private Information Retrieval (PIR)[58] could in principle be implemented to obfuscate which client has interest in exactly what information, once an EPCIS has been located.

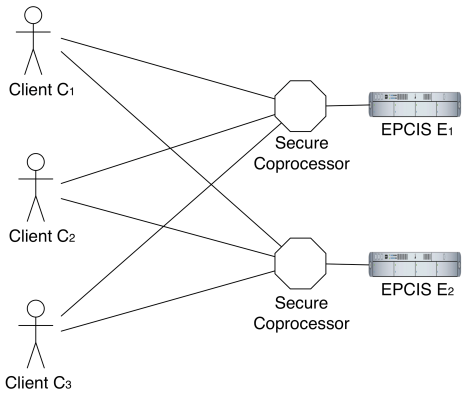


Figure 4.13: PIR for EPCIS Access

Fig. 4.13 shows a conceptual picture of a PIR system based on secure coprocessors (SC), which manage the database access including access control implemented in the SC, but obfuscate the client interest in specific data against the EPCIS provider.[59]

But in the case of a globally distributed IOT name service like ONS, problems of scalability, key management, and monetary cost in case of additional secure hardware, as well as general performance issues seem to render PIR approaches impractical.

[57] Wright et al., 2004 [216]. For a in-depth discussion of side channel analysis and unintended information leakage not only in Tor cf. Murdoch, 2007 [135].

[58] Chor et al., 1998 [31]; Kesdogan et al., 2003 [109]; Asonov and Freytag, 2002 [6]; Berthold, 2005 [15]; Iliev and Smith, 2005 [93].

[59] This example system is an application of the Private Database Access (PDA) method by Berthold, 2005 [15], to EPCIS, and is subject to ongoing joint research with O. Berthold and S. Gürses on multilateral EPCIS security.

4.5 Summary

In the previous sections we discussed MONS, the multipolar redesign of the ONS Root, and further potential measures to secure parts of the EPCglobal Network. For a preliminary and very general evaluation of their effect and practicality, see Table 2, where also the discussion of Peer-to-Peer (P2P) systems of the next Chapter 5 is reflected. Some of these methods could also be combined to create alternatives to the current EPCglobal Network design.

Countermeasure	Anonymity	Confidentiality	Integrity	Availability	ONS?	EPCIS?
MONS				+	++	n/a
ONSSEC			++		+	n/a
VPN		++	++		– –	– –
SSL, TLS		++	++		– –	++
Mixes, OR	++	+			–	+
PIR	++	++		–	– –	–
P2P	+			++	++	+

Table 4.1: Practicality of Countermeasures

We assume the actual EPCIS communication to be more easily securable against third parties than ONS, for example by using standard authentication and encryption by TLS – though integrity problems through improper certificate handling might spoil this assumption, and availability problems do occur likewise. In addition, every single EPCIS constitutes an attractive opportunity for query data analysis, but only for observing a specific user segment of the EPCglobal Network (i.e., the customer base of a specific company), unlike, for example, the global scope of the ONS Root.

If ONS is based on DNS as has been proposed in its specification, a whole new branch of privacy problems do arise, which could only in part be mitigated by current security technology, and would even then require huge efforts in network design. For companies and individuals alike, traffic anonymizers like Tor could present an interesting partial solution to privacy-preserving ONS use and EPCIS access. This approach should be investigated further in relation to scalability, manageability, and adverse effects on possible authentication measures for accessing the EPCIS.

Integrity of ONS information could be achieved by deploying DNSSEC, though this needs to be set up between all possible business partners and information service providers, which seems very unlikely given the current diverse and complex state of the Internet. Availability of ONS and EPCIS servers is a problem that would have to be approached and dealt with by every company in the resolution path.

Moving from barcode to RFID tags containing an EPC was motivated by saving costs and simplifying supply chains, without taking confidentiality concerns of individuals or companies acting as clients into account. The implementation of a global system to store and access heterogeneous information about products appears likewise at least in part be motivated by future after sale business. Again, security and privacy measures are no integral part of the original design, but – if at all – an

afterthought. Based on a deeper analysis of the multilateral requirements reflecting the security interests of all stakeholders involved, there is urgent need to design an alternative model to ONS along with protocols for its implementation, and to avoid similar pitfalls for Discovery Services.

One promising research direction is the use of Peer-to-Peer systems based on Distributed Hash Tables (DHT) for ONS resolution, which will be presented in the next chapter.

Chapter 5

Paradigm Shift: P2P-ONS

5.1 Introduction

Highly distributed alternatives to classical network service architectures exist in the form of Peer-to-Peer Systems (P2P), which can be considered to be a paradigm shift from the classical client–server architecture to a new paradigm with a roughly equal distribution of responsibility and load among peers.

Especially structured P2P systems using Distributed Hash Tables (DHT) offer high robustness to faults, avoid single points of failures (e.g., they have no single root like DNS), and distribute responsibility and load among participants in a systematic way by means of a prearranged topological overlay structure.[1] In light of ongoing projects like *Cooperative Domain Name System* (CoDoNS)[2] that build viable alternatives to classical DNS, it seems reasonable to assume that an IOT name service like ONS could also be based on a DHT architecture.[3] Simply switching ONS to a P2P architecture, however, would not guarantee integrity and confidentiality, though it would enhance anonymity in practice by avoiding a single ONS Root, and by increasing the number of nodes an adversary would have to monitor for incoming ONS requests. Other attack vectors for intercepting EPCs from clear text Internet traffic would be possible without further countermeasures.

In this chapter we present OIDA, the *Object-Information Distribution Architecture*, which is an alternative to ONS based on DHT.[4] OIDA started out from the idea not to let an object identifier, for example an EPC, cross the Internet in clear text, but to use a cryptographic hash value instead. This initial idea was combined with the flat identifier space and uniform node and record distribution of a DHT, which is also

[1] Balakrishnan et al., 2003 [10]. An excellent collection of introductory articles on P2P systems is given in Steinmetz and Wehrle, 2005 [193].

[2] Ramasubramanian and Sirer, 2004 [161]. URL: `http://www.cs.cornell.edu/people/egs/beehive/codons.php` (05.2008)

[3] For a discussion of possible latency penalties see Section 5.5.3.

[4] This chapter is extending previous work published in Fabian and Günther, 2007 [57].

achieved by using suitable cryptographic hash functions. OIDA was implemented as a prototype on PlanetLab, an international network research platform. The basic architecture provides high performance, and can be extended by additional security measures to achieve a balance between better security and performance overhead.

The following research areas are related to the topic of this chapter.

ONS and Discovery Services. The main established proposal for an IOTNS is the Object Naming Service (ONS), which was presented and discussed in Chapter 3. In addition, several groups are currently working on Discovery Services (DS), besides EPCglobal for example the EU project BRIDGE[5] (under the coordination of GS1), and an emerging IETF working group. At present, the final definition and requirements on Discovery Services are not clear, yet. Proposals range from an extended serial-level IOTNS to more complex middleware layers.

P2P-DNS. To improve DNS robustness and performance, several designs using P2P systems have been proposed.[6] The important Cooperative Domain Name System (CoDoNS),[7] for example, combines the decentralization, scalability, simple administration, and robustness of a DHT with proactive caching to reduce lookup latency; it offers data authentication based on cryptographic delegation and DNSSEC. Access control for the data stored in the DHT is not provided, as there is no encryption offered by any of those designs. Consequently, the DHT could not easily be used to store confidential item or address data. Likewise, client confidentiality requirements are not fulfilled. CoDoNS is also a clear text protocol like DNS.

Cooperative DNS lookups (CoDNS),[8] and also hybrid DNS and DHT architectures have been proposed in the literature.[9] But so far, all those approaches did not foresee or consider the privacy and security issues caused by using DNS in the context of RFID and the IOT.

Local Discovery Protocols, UDDI. Many protocols exist that provide device and service discovery in small-scale and often local networks.[10] Most of them will not scale to the global environment necessary for an IOTNS. In particular, Universal Description, Discovery, and Integration (UDDI) is a Web service standard for service description and discovery, but is currently mostly used within a single organization, lacking an established global infrastructure so far. UDDI helps to discover services of

[5] `http://www.bridge-project.eu/` (05.2008).

[6] See e.g. Cox et al., 2002 [37]; Ramasubramanian and Sirer, 2004 [161]; Doi, 2005 [46]; 2007, Huang, [91].

[7] Ramasubramanian and Sirer, 2004 [161].

[8] Park et al., 2004 [151]. Cf. also ConfiDNS Poole and Pai, 2006 [159].

[9] Balakrishnan et al., 2004 [11]; Doi, 2005 [46].

[10] Wikipedia, s.v. *Service Discovery*, [215] (05.2008).

a specific, rather high-level, functionality, but would probably not scale for looking up vast numbers of OIDs.[11]

Privacy-Enhancing Technologies (PET). There is much past and present work on enhancing the privacy of users of network and service infrastructures. Important approaches include mix networks and private database access. Mix networks and onion routing systems like Tor[12] are general-purpose systems that offer a high degree of anonymity for its users (cf. Section 4.4.3).

Freenet[13] is an anti-censorship system that even combines elements of mix networks and P2P systems. But DHT-based P2P systems promise better performance as a look-up service than mixes or even onion routing because the latter require extensive cryptographic operations on intermediate nodes. Yet, more extensive studies of this performance vs. anonymity trade-off – using different and realistic traffic patterns – must be conducted before a final conclusion can be reached. Methods for PIR could be adopted for EPCIS access, but seem yet to lack scalability and performance for use in global and dynamic lookup services (cf. Section 4.4.4).

Distributed Storage Networks. There is also much research on distributed storage and content delivery networks (CDN), including their anonymity and censor-resistance.[14] Those systems include the Eternity Service,[15] FreeHaven,[16] OceanStore,[17] Cooperative File System,[18] and Publius.[19] Some of these systems could potentially be able to work as a replacement for ONS, Discovery Services, EPCIS – and even the whole EPCglobal Network. The main problem – besides unclear performance characteristics and scalability – is the possible lack of acceptance on the information provider's side to "let go of their information" and store it somewhere in untrusted systems, outside corporate boundaries. The same problem would occur with DHT, and for this reason we think it practical to keep the two separate phases of the original design (i.e., IOT name service lookup and EPCIS access). If those objections should not hold in future, distributed storage systems – which are often based on P2P architectures – with client anonymity and access control could provide interesting alternatives to the EPCglobal Network as a whole.

---

[11] Cf. A. Rezafard, Extensible Supply-chain Discovery Service Problem Statement, IETF draft (work in progress), `http://www.ietf.org/internet-drafts/draft-rezafard-esds-problem-statement-01.txt` (05.2008).

[12] Dingledine et al., 2004, [45].

[13] Clarke et al., 2002 [33]. A major revision of the Freenet software appeared in 2008, `http://freenetproject.org/` (05.2008).

[14] For a survey cf. Androutsellis-Theotokis and Spinellis, 2004 [4].

[15] Anderson, 1996 [2].

[16] Dingledine et al., 2001 [44].

[17] Rhea et al., 2001 [167].

[18] Dabek et al., 2001[39].

[19] Waldman et al., 2000 [205].

This chapter is structured as follows. First, we discuss DHT fundamentals. Then we present the OIDA architecture including results from deploying and testing a prototype on PlanetLab.[20] In addition, we conduct a feasibility and security analysis of OIDA, and discuss possible future extensions to cope with remaining risks. A summarizing comparison of ONS, MONS, and OIDA concludes this chapter.

5.2 Distributed Hash Tables

In this section, the basic concepts of Distributed Hash Tables (DHT) are presented. The general advantages of DHTs will be discussed, and a short overview on specific DHTs and applications will be given.

Distributed Hash Tables are P2P systems that offer a lookup functionality analogous to a hash table, but in a distributed and decentralized fashion, involving multiple computers without central control.[21] DHT offer a simple lookup and storage interface based on a one-to-one correspondence between data items and keys. The underlying distributed DHT algorithms determine which nodes are responsible for storing the data by organizing keys and nodes in a logical *overlay network*, which is in general independent of the physical or IP network topology on lower layers (see Fig. 5.1), using concepts like consistent hashing[22] with only few local information about the whole system. Consistent hashing balances data items to nodes in a roughly uniform way, and allows for node joining and leaving, without the need for major redistribution of keys and data in the running system.

Most DHTs resolve lookups in $O(\log N)$ hops through the overlay network, where N is the number nodes in the DHT, which offers excellent scalability. This feature connects DHTs with recent general research on complex networks, especially *small-world* networks, in which the average path length scales at the most logarithmically with N.[23]

This scalability is enhanced by the fact that the routing table size and amount of state information stored at any particular node also scales with $O(\log N)$, which means that every node just needs to know a very small part of the whole overlay graph.

DHTs also offer functionality like message routing in the overlay, node joining and leaving procedures, and data redundancy in a self-organized fashion. Particular DHTs use several different algorithms[24] and *overlay topologies*[25] to structure node

[20] Peterson and Roscoe, 2006 [155]. More on PlanetLab in Section 5.4.

[21] Ratnasamy et al., 2001 [164]; Balakrishnan et al., 2003 [10]; Steinmetz and Wehrle, 2005 [193]; Wikipedia s.v. *Distributed Hash Table*, [215] (05.2008).

[22] Karger et al., 1997 [106]; Stoica et al., 2003 [197].

[23] Watts; 1999 [210]; Kleinberg, 2000 [112]; Loguinov et al., 2005 [124]; Thadakamalla et al., 2007 [200].

[24] For an overview cf. Ghodsi, 2006 [75].

[25] Also called *overlay geometries*. A comparison is given in Gummadi et al., 2003 [80].

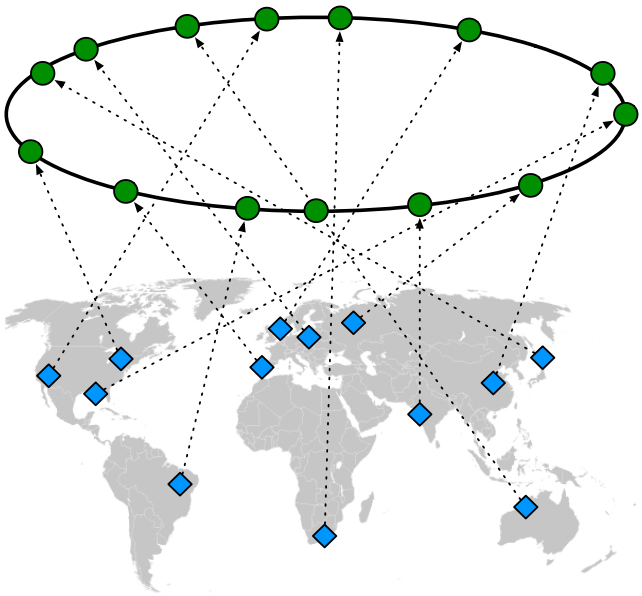


Figure 5.1: DHT Overlay vs. Physical Topology

identifiers and keys.[26] In addition, they differ in the amount of performance enhancements they offer, and in the maturity of code available so far. Example DHTs include *Chord*,[27] Pastry, and Bamboo, which use circular identifier spaces, *Koorde*,[28] based on De Bruijn Graphs, and *CAN*[29] that uses a virtual multi-dimensional torus.[30]

P2P systems in general have proven their scalability and performance in real-world applications. Structured P2P systems based on Distributed Hash Tables (DHT) in particular are the foundation for several distributed network services that already run in reality, e.g., on international testbeds for future network services, such as PlanetLab. Some of those applications are name services, e.g., DNS alternatives or other resource lookup services, like P2P-SIP.[31]

In the following, we present OIDA, an example P2P IOT name service architecture, specifically for P2P-ONS.[32]

[26] For this reason DHTs are often called *structured* P2P systems, in contrast to *unstructured* designs like *Gnutella*.

[27] Stoica et al., 2001 [196]; Stoica et al., 2003 [197].

[28] Kaashoek and Karger, 2003 [102]

[29] Ratnasamy et al., 2001 [164].

[30] Noteworthy DHT designs which can only be mentioned here include *Tapestry*, *P-Grid*, *Kademlia*, *Symphony*, *Viceroy*, *Cycloid*, amongst others.

[31] DNS based on DHT will be discussed in Section 5.5.3. For P2P-SIP projects see: `http://www.p2psip.org/implementations.php` (05.2008). Also cf. Baumgart, 2008 [13].

[32] We create this term to have a practical name for an *architecture category*, in line with established names like P2P-SIP and P2P-DNS, not aware of any already existing use.

5.3 OIDA

In this section, a DHT-based IOT name service architecture called *Object Information Distribution Architecture* (OIDA) will be discussed. OIDA involves the following key ideas: Each interested company deploys dedicated OIDA-Nodes. Those nodes form an overlay network using an ID space specific to the DHT in use, where a cryptographic hash function maps EPCs and nodes to overlay IDs. This pseudo-random mapping of identifiers to storage nodes balances load more evenly, allows for easy replication, avoids single points of failure, and reduces the feasibility of targeted attacks against specific information providers or clients. The DHT provides the routing to the responsible nodes, as well as joining, leaving, repair, and optimization procedures, without a central entity managing those operations.

Nodes store deterministically assigned – but from the perspective of a node owner or adversary interested in specific EPCs, apparently random – encrypted and signed documents belonging to hash value ranges. Those documents may contain object data or EPCIS IP addresses, because if possible indirect use of DNS should also be avoided for reason discussed in Chapter 3. For scalable data authenticity, the existence of a certification authority (CA) infrastructure is assumed,[33] which can also be distributed, similar to multipolar ONSSEC (see Section 4.3.3), or a web of trust.[34]

5.3.1 Cryptographic Hash Functions

Before we present OIDA in detail, we investigate one central element of DHT designs from a different perspective than usually adopted in the DHT literature.

The *hash functions* used in DHTs for uniform distribution of pre-images (e.g., node and data names) to the overlay identifier space are in general stronger than theoretically necessary for this application. In most implementations, they are a reuse of established building blocks for network security applications: so-called *cryptographic* hash functions, which fulfill more cryptographic requirements than the nearly uniform output distribution.[35]

Definition 3. *A* Cryptographic Hash Function *(CHF) is a deterministic function h mapping a bit string of arbitrary length to a hashed value of fixed length, with the following desired properties:*

1. *(Nearly) uniform output distribution.*[36]

[33] The use of X.509-based PKI (RFCs 3280, 5280) is planned for the EPCglobal Network EPCglobal, 2008 [54].

[34] Zimmermann, 1995 [218].

[35] Mao, 2004 pp. 300 [127], also for the following definition.

[36] More precise: On any input x, the output $h(x)$ should be computationally (polynomial-time)

2. *Collision resistance: It should be computationally infeasible to find x and y, $x \neq y$, s.t. $h(x) = h(y)$.*

3. *Pre-image resistance (also called one-way property): Given a hashed value h, it should be infeasible to find an input string x s.t. $h(x) = h$.*

4. *Practical efficiency: $h(x)$ should be easy to compute.*

Property (4.) benefits DHT lookup and storage performance for clients and information providers, property (1.) guarantees the nearly uniform output distribution of keys for large input spaces, providing a balancing of responsibility, risk, and load. Property (2.) helps with functional correctness of the service, that is, documents for different names should be stored under different overlay keys (with high probability). But property (3.) will turn out interesting for EPC confidentiality in OIDA. While using a CHF for DHT hashing, it should be infeasible to calculate a pre-image EPC e from a stored or transmitted value $h(e)$.

One prominent example for a CHF is SHA-1,[37] but there are recent cryptographic results which question its security with respect to collision resistance.[38] But so far this does seem not to be affecting pre-image resistance, nor a useful but weaker property than collision resistance, *2nd pre-image resistance*.[39] According to NIST, however, developers and federal agencies in the US should soon switch to the SHA-2 family of CHFs.[40]

In conclusion, the use of SHA-1 today as a hash function to generate DHT overlay IDs can still be considered secure with respect to pre-image resistance, but future systems should switch to the SHA-2 family to avoid surprises by further research in SHA-1 security.[41]

5.3.2 OIDA Architecture

In this section, we present OIDA as a general DHT-based architecture for ONS at a conceptual level. Our point is to analyze if and how access control could be

indistinguishable from a uniform binary string in the output interval, cf. Mao, 2004 pp. 131, 300 [127]. The existence of such functions is based on a plausible assumption in complexity theory, ibidem, p. 132.

[37] Cf. RFC 3174, Eastlake and Jones, 2001 [48]; NIST, 1993, [140].

[38] Wang et al., 2005 [209], reduced the average attack effort from 2^{80} operations for brute force collision search to less than 2^{69} operations, which could be feasible for determined attackers today, see also Burr, 2006 [26].

[39] It is computationally infeasible to find any second input y which has the same output $h(y) = h(x)$ as any *already given and fixed* input x, cf. Menezes et al., 1997, pp. 323 – 325 [131], where also alternate definitions for hash functions are presented.

[40] Burr, 2006, p. 91 [26], `http://csrc.nist.gov/groups/ST/hash/statement.html` (05.2008). A description of SHA-224 is given in RFC 3874, an analysis of the whole SHA-2 family in Gilbert and Handschuh, 2004 [77].

[41] Though the SHA-2 family is related to SHA-1, according to NIST similar attacks seem infeasible well beyond the year 2010. Nonetheless, research into new CHFs was initiated.

provided and client privacy be enhanced compared to ONS, while keeping the main DHT functionality unchanged. In OIDA, information providers publish address documents to the DHT for single EPCs or whole EPC classes, a possible convention for the latter could be to set the serial part to zero. These documents contain the address lists of corresponding information servers (EPCIS) that are queried for by OIDA clients, or even parts of the object information itself.

To keep the decentralized and self-organized aspects of P2P systems, we do not yet demand security features on the nodes themselves in the basic OIDA architecture, except for the ability to verify the identity of information providers. Nodes should not need to be more trustworthy from a client's perspective than unauthenticated DNS servers used daily on the Internet. This initial design choice is made to investigate the limits of untrusted P2P systems, with their advantage of self-organization and low computational overhead. In a real implementation of OIDA as an infrastructure network, for example formed by those special corporate hosts already designated to work as ONS servers, this choice could be lifted easily, and additional inter-node security measures may be implemented, possibly in conjunction with incentive and reputation systems in the case of a more open membership.[42] For now, most security features in the basic OIDA architecture are put into the stored documents.[43]

The cryptographic hash value $h(e)$ of an EPC e plays two important roles: first, as a DHT lookup key, second – due the one-way property, as will be discussed later in depth in Section 5.6.5 – as a confidentiality-enhancing measure to avoid sending the EPC in clear text across third party networks. To increase the strength of this privacy aspect of the protocol, it would be helpful if the information provider and client share an additional common value s, which could then be used as a *salt* for the CHF. We discuss its function and particular requirements in Section 5.6.5. If such a salt s is unfeasible due to lack of secure distribution channels, or is unwanted because completely unrestricted and most flexible lookup of data is required, s can be assumed to be the empty string in the following.

An information provider P who likes to publish information i for a given EPC e – e.g., the address of a corresponding EPCIS – first creates a document containing its name P, the information i, and – for detecting authenticated, but wrongly assigned messages – the cryptographic hash of the concatenation of e and s. Additionally, version control information, time stamps, and TTL values should be included. If a central CA is used for OIDA, a certificate signed by it could be added, linking P with its public key P_{pub}.

To implement access control and to reduce the risk of inference attacks from the data included in the returned document, this data should be encrypted, for example by using a shared key k and a symmetric cipher like AES.[44] Prerequisites for this

[42] Fischmann, 2006 [66].
[43] In Section 5.6.2 we will discuss special circumstances where a node PKI would be beneficial.
[44] For AES cf. NIST, 2001 [141].

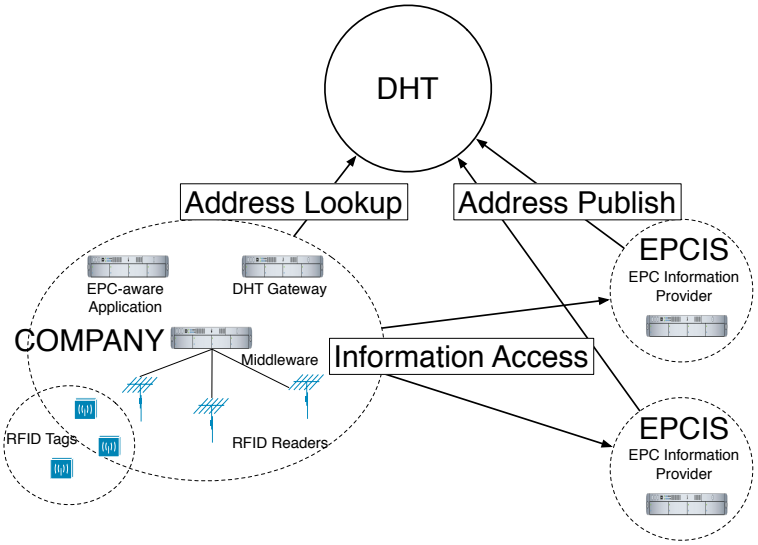


Figure 5.2: OIDA Architecture

are discussed in Section 5.6.5. P signs a CHF value – for storage efficiency – of this document by using his private key P_{priv}, and adds this as a signature.

This storage step could include identification and authentication of P by the responsible nodes to avoid spam, mutual authentication for enhanced security, and additional replication to increase availability. In the basic architecture, we simply demand publisher-controllable redundancy of data storage to avoid single points of failure, easily achievable by a convention for the CHF input using a replica identifier r, see Section 5.6.2.

The final document d is then stored r_{max} times in the DHT at the nodes responsible for overlay IDs $h(s, e, r)$, $1 \leq r \leq r_{max}$, by contacting a DHT node acting as a client gateway, for example situated in the manufacturer company itself (see Address Publish in Fig. 5.2).

At a later time, for example once the item corresponding to the EPC has been acquired by the OIDA client C who is now in the possession of EPC e and salt s, C requests information about e by issuing a request for $h(s, e, r)$ to one or many DHT gateways, using arbitrarily many values $r \in \{1, ..., r_{max}\}$, but at least until a copy of the document d is successfully retrieved. The DHT replies to those requests by sending d – possibly multiple times – via the gateways to C (see Address Lookup in Fig. 5.2).

C then decrypts d, hashes it, and verifies the hashing result to the signature after having applied decryption using P_{pub} to it, in order to determine if P really created this document and to verify the integrity of d – if any problem occurs, C requests another replica by varying r. This verification procedure may include the investigation of the public key and the certificate binding it to P, signed by the CA, which should also be trusted by C – both elements are retrieved from the document

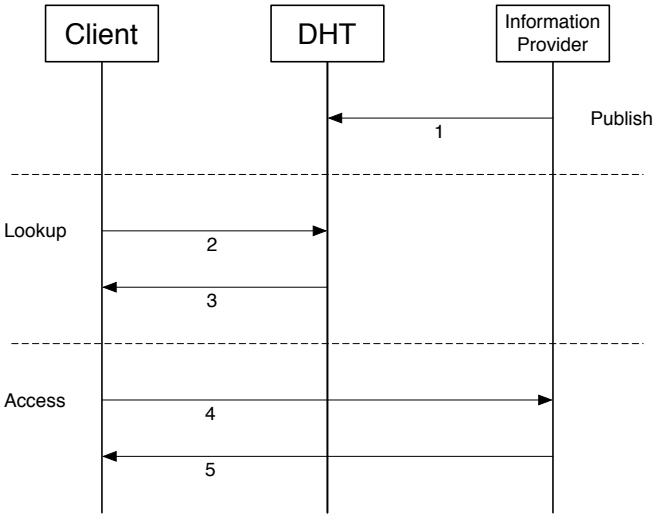


Figure 5.3: OIDA Protocol

interior after decryption.

If C trusts the certificate and likes to fetch information from P, a direct connection – e.g., via Web services – to the EPCIS address i stated by P is established (see Information Access in Fig. 5.2). This EPCIS access can be authenticated and encrypted, e.g. by Transport Layer Security.[45]

At the conceptual layer, the basic OIDA protocol therefore works as follows (see Fig. 5.3):

1. Publish: $S \longrightarrow DHT$: `dht-store`$(h(s, e, r), d)$.

2. Lookup Request: $C \longrightarrow DHT$: `dht-retrieve`$(h(s, e, r))$.

3. Lookup Reply: $DHT \longrightarrow C$: d.

4. EPCIS Access: C verifies d, and if genuine, starts a request to the EPCIS of P, located at address i extracted from d, using TLS.

5. EPCIS reply from P to C, using TLS.

After this conceptual presentation of OIDA, we now turn to a discussion of organizational aspects of OIDA deployment.

[45] TLS, Dierks and Rescorla, 2006, [43], possibly enhanced by extensions for better client authentication, cf. Oppliger et al., 2008 [145].

5.3.3 Organizational Aspects

OIDA and a Central Entity

There are several organizational aspects of using OIDA as an IOTNS. To guarantee liability, especially in business environments using the IOT, there should be a common agreement and procedure in place that assigns EPC ranges to authoritative entities who are allowed to publish address records. An example convention would be that EPC Managers may publish address data exactly for EPCs belonging to their own company as would be indicated by the company prefix. Similar policies could be created for other object identifier systems.[46]

Another question would be, who are the clients who may use the IOT name service? Is it publicly accessible, or restricted? OIDA, for example, could be queried by everyone, but only clients authorized by the information publisher – via document encryption, and key management and access control policies – could decrypt the returned address information. At the EPCIS tier, conventional Web service security measures would enforce the publishers security policy. One model to achieve this could be a central registry that issues identifier ranges and verifies the identities for authoritative entities, for example GS1 and EPCglobal in the case of the EPC and EPCglobal Network. However, this central position is very powerful, controlling – and possibly also denying – the access of third party information providers to the IOTNS.

A central entity should also monitor the uptime of OIDA nodes, issue software updates, and regulate data storage in such a way that every publisher provides at least enough nodes and storage capacity as needed for the data amounts he wants to store in OIDA. All of this could be part of detailed business contracts.

For key and certificate management to be scalable, there should be some kind of public-key infrastructure in place, which may be hierarchical – having a power center potentially suffering from unipolarity or other misuse – or a web of trust. This could also be tree-like structure with a distributed root, like Multipolar ONSSEC (see Section 4.3.3).

To sum up, even with a P2P system like OIDA there could be some important management role to be played by a central entity like EPCglobal. Alternate organizational models, however, could be investigated for every function such a central entity would adopt.

Document Versioning, Update, and Revocation

Back to a more operational perspective, the address documents published to OIDA could correspond to arbitrary object-identifier frameworks, including barcodes. In

[46] Therefore, fulfilling the membership and authorization requirement (cf. Section 2.2) in OIDA would be the task of the *environment*, outside of the scope of the *machine*, similar to ONS.

the case of an EPC, the full EPC including serial number may be the (pre-image) lookup key to enable serial-level document retrieval. Also only partial EPCs for whole object classes, or only the EPC Manager could be used, for example to retrieve general manufacturer links, or as will be discussed in Section 5.6.5, lists of salts.

If serial-level information is used, but the EPCIS addresses do not change for different items of the same class, a corresponding class-level entry could also be included in the response. This can be cached locally (within the trusted network bounds of an organization), so that new queries for objects of the same class are answered directly from the local cache.

The documents should include a version number and possibly time-stamps, to let the client decide between different versions of the document issued at different times,[47] in case the replication process was incomplete due to a node error, or a document update is currently taking place. A time to live (TTL) field could indicate – as in DNS – how long the data should be considered valid and can be kept in local caches, before a new copy should be requested from OIDA.

In addition to a `dht-store` and `dht-retrieve` API, most DHTs also offer interfaces to update or delete records. Those should offer access control to restrict them to the authorized publishers that created the records.

Integration Into Enterprise Networks

OIDA nodes should be placed in a corporate Demilitarized Zone (DMZ), as with other servers accessible from the Internet. Internal OIDA proxies should be used to concentrate OIDA queries of internal clients. Those proxies could also be used for translating ONS request to OIDA queries. They should be able to contact the corporate OIDA nodes in the DMZ for queries and publishing, and should for redundancy also be able to query foreign OIDA machines.

After this conceptual discussion, an implementation of OIDA is presented in the next section.

5.4 OIDA Prototype

In the following, we present an OIDA prototype implemented on PlanetLab, which will be introduced next. The Bamboo DHT and details of the prototype are then described. A set of experimental results concludes this section.

[47] Multiple documents for any ID can be stored, of which all or selected subsets could be retrieved.

5.4.1 PlanetLab

PlanetLab[48] (PL) is an international research network for the development of new network services. For a description of its history see Fiuczynski, 2006 [67], for design principles confer to Peterson and Roscoe, 2006 [155], and for experimental system research on PL in general, Peterson and Pai, 2007 [154] can be consulted.

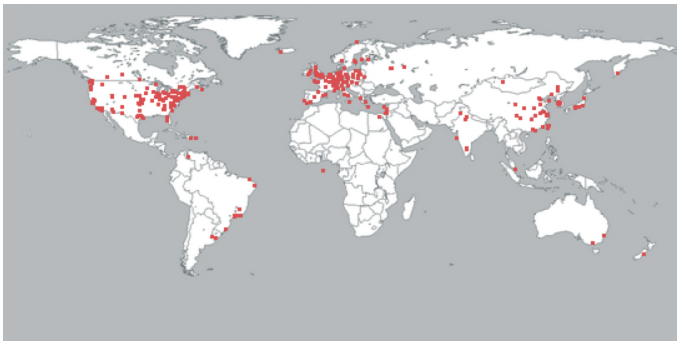


Figure 5.4: Geographical Distribution of PlanetLab

Virtual hosts on PL nodes can be reserved for projects and assigned to host groups (*slices*) under exclusive control of one experimenter, but the actual physical nodes must be shared with dozens of concurrent experiments at a given time. Therefore, PL offers a real world testbed under realistic load, but cannot guarantee that experiments are exactly reproducible at a later time. In our experience, however, the general quality level of the results remained quite stable.

PlanetLab consisted of around 850 nodes at 428 sites in April, 2008 (Fig. 5.4). Of those, mostly by automatic selection via timeouts, our experiments used roughly 350 nodes at a given time, mostly stable nodes with long uptime, and a better network connection to our testing clients to avoid timeouts.

5.4.2 Bamboo DHT

Our OIDA prototype is based on the Bamboo DHT, mainly because of its relatively mature status,[49] and due to its design goal of withstanding *churn*, that is, frequent change in membership due to ongoing node departures, failures, and arrivals, a property we deemed important for a prototype using the experimental platform PlanetLab.

For a more mature version of OIDA, deployed as a production infrastructure network, business contracts should guarantee a more stable node membership, but the ability to handle churn would be a plus for service robustness. Bamboo is described

[48] URL: `https://www.planet-lab.org/` (04.2008).
[49] Bamboo is used for Open DHT, cf. Rhea et al., 2005 [169], a long-running DHT on PlanetLab.

in Rhea et al., 2004 [168].[50]

ID Space and Routing

Bamboo has evolved from the Pastry DHT[51] and inherits its overlay geometry (a circle) and routing mechanisms. However, a larger identifier space $[0, ..., 2^{160})$ of cardinality 2^{160} is used, corresponding to the possible output space of the SHA-1 CHF,[52] which is used in Bamboo for creating an overlay ID from a pre-image, which is a node's (IP address, port) tuple, or a data identifier.

Routing in Pastry and Bamboo uses two main sets of state information that have to be maintained by each node: The first is the *leaf set* L for connections to the k preceding and k subsequent nodes in the ID circle[53]; let $U(L)$ denote the corresponding interval of the overlay ID space.

The second set is the *routing table* for larger hops through the ID space – similar to the *finger table* in Chord. The routing table of a node A (with overlay ID a) contains nodes whose overlay IDs share successively longer prefixes with the overlay ID of A, where each ID is regarded as sequence of digits with base 2^b, and b is a fixed parameter of the deployed DHT. The routing table consists of $\frac{160}{b}$ rows and $2^b - 1$ columns, but is in general not completely filled. If available, an entry $R(i, j)$ of row i and column j should contain the IP address of a node whose identifier matches that of A in exactly i digits and whose $(i + 1)$th digit is j.

It is possible that no such node is known, leaving the entry empty, or that multiple candidates exist, in which case one of them is chosen according to a specific metric, for example in Bamboo proximity in the network topology.[54] On average, for a network of N nodes, only $\log_{2^b} N$ routing table rows are populated.

The routing procedure works as follows (cf. Algorithm 1). If A likes to route a message M to a node responsible for destination key d, first the leaf set is consulted: If d is an element of the corresponding interval, M is forwarded to the numerically closest node L_i in the leaf set (mod 2^{160}). Otherwise, the length i of the longest matching prefix of d and a is computed, and the routing table entry $R(i, d(i + 1))$ is consulted, where $d(i + 1)$ denotes the $(i + 1)$th digit of d. If this entry exists, M is forwarded to the corresponding node.

---

[50] The Bamboo source code is available from `http://www.bamboo-dht.org` (04.2008).

[51] Rowstron and Druschel, 2001 [176].

[52] Cf. RFC 3174, Eastlake and Jones, 2001 [48]; NIST, 1993, [140].

[53] This set is comparable in function to the set of *successors* in Chord, see Stoica et al., 2003 [197]. Often, in general DHT studies, the term set of *sequential neighbors* is used, cf. Gummadi et al., 2003 [80].

[54] This procedure is called Proximity Neighbor Selection (PNS), see Rhea et al. 2004 [168]. Other methods, which are like PNS also usable for other DHTs like Chord, include: Deterministic, Random (RNS), and Long Lifetime Neighbor Selection (LNS), which is based on the uptime of nodes. Of those, LNS was shown to provide better performance under churn by Zhu and Yang, 2006 [217].

Otherwise, M is routed to a node known in any table, numerically closer to D, which shares the same prefix with d as a does in Pastry,[55] or in Bamboo to another node of the leaf set numerically closest to d.[56] In an ideal network, the leaf set guarantees *routing correctness*, because by construction a numerically closer node must exist.

But also in a network under heavy churn message delivery is guaranteed unless $\frac{|L|}{2}$ or more nodes with consecutive IDs fail simultaneously; a very improbable case due to the anticipated geographic and organizational diversity of corresponding nodes, but with a possible worst case of a number of routing steps linear in N.[57] On average, however, assuming accurate routing tables, the expected number of routing hops is $O(\log_{2^b} N)$.[58] Keeping the routing tables accurate to achieve this excellent scalability is therefore a major task for DHT self-organization.

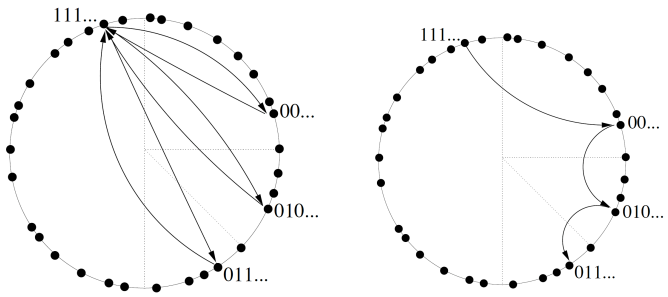
Algorithm 1: Routing in Bamboo

```
if d ∈ U(L) then
    Next-Hop ← L_i ∈ L s.t. |d − L_i| min. #L_i is already final hop.
else
    if R(i, d(i+1)) is not NULL then
        Next-Hop ← R(i, d(i+1))
    else
        Next-Hop ← L_i ∈ L s.t. |d − L_i| min. #L_i is only next hop.
```

Like other DHTs, e.g., Chord – and also similar to DNS query modes – Bamboo supports two different query routing modes: *iterative* and *recursive* routing. With iterative routing (Fig. 5.5(a)[59]), there is only one source A that issues all queries, each intermediate hop only returns the address of the next hop H to A, which then contacts H by itself. Recursive routing (Fig. 5.5(b)) forwards the message itself from hop to hop, and is the default mode in most DHTs. A possible answer could take the same route back or may be delivered directly by using an embedded address of the query source.

If a DHT is used for enhancing anonymity of a name service, only recursive routing with indirect answer delivery should be used, otherwise the destination and every hop in-between could see the source of the query, and the anonymity situation would be equivalent to ONS (cf. Section 3.4.3).

[55] Rowstron and Druschel, 2001, p. 5 [176].
[56] Rhea et al., 2004, p. 3 [168].
[57] Rowstron and Druschel, 2001 [176].
[58] Ibidem.
[59] Both Figures are cited from Rhea et al., 2004 p. 3 [168].



(a) Iterative Routing in Bamboo (b) Recursive Routing

Figure 5.5: Iterative and Recursive Routing (Source: Rhea et al., 2004)

Node Joining and Failure

It is necessary that a new node A joining Bamboo or Pastry knows at least one existing node B of the DHT. A sends B a special *join* message to its own overlay key a (corresponding to A), constructed from the SHA-1 hash value of (IP, Port) of the Bamboo installation at A – other ways to generate node IDs in a unique way could for example use public keys. This join message is routed to the numerically closest node Z. All nodes on the path send their state tables to A, which uses this information to build its own state table, inform other nodes of its presence, and to become responsible for a part of the ID space.[60]

Node failures are detected during routing, and proactively by periodically checking the liveness of neighboring nodes.[61] In summary, node joining and failure can be achieved without central coordination, without huge or global changes in the ID space assignment, and in a self-organized fashion, reducing management overhead compared to DNS.

5.4.3 Prototype Details

The OIDA prototype (Fig. 5.6) consisted of the Bamboo DHT, as well as client scripts to encrypt, sign, store, retrieve, and verify data from several machines outside of PL, however, without implementing a truly global CA issuing certificates. Direct signature verification was used, trusting in the correctness of a publishers public key, because the deployment of a real CA and trust hierarchy was considered to be part of established network engineering, outside of the scope of the prototype.

Bamboo was deployed in a dedicated PL slice on more than 350 nodes, distributed over all continents. For a pictorial snapshot of the overlay ring structure see Fig.

[60] For details cf. Rowstron and Druschel, 2001 pp. 7 [176].

[61] Repair procedures are described in Rowstron and Druschel, 2001, pp. 8 [176], as well as in Rhea et al., 2004, p. 6 [168], where the periodic recovery of Bamboo is described and shown to save bandwith in face of churn.

5.7(a), and Fig. 5.7(b) shows a snapshot of all nodes leaf sets (at the circle perimeter) and routing tables (circle interior), visualizing the structural robustness of the DHT that is achieved even without a fully connected graph.

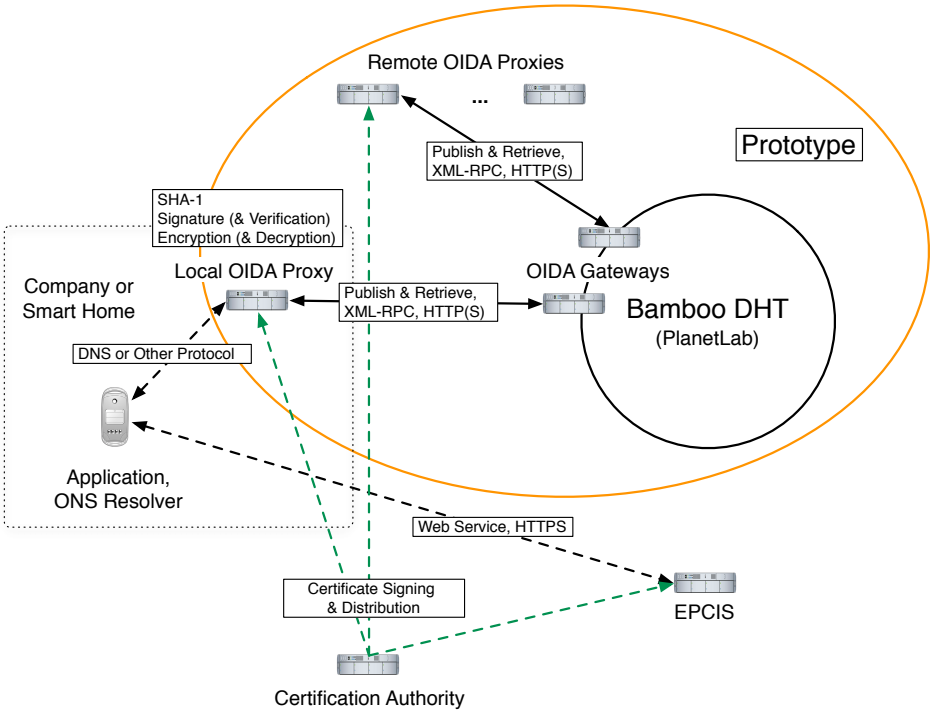


Figure 5.6: OIDA Prototype on PlanetLab

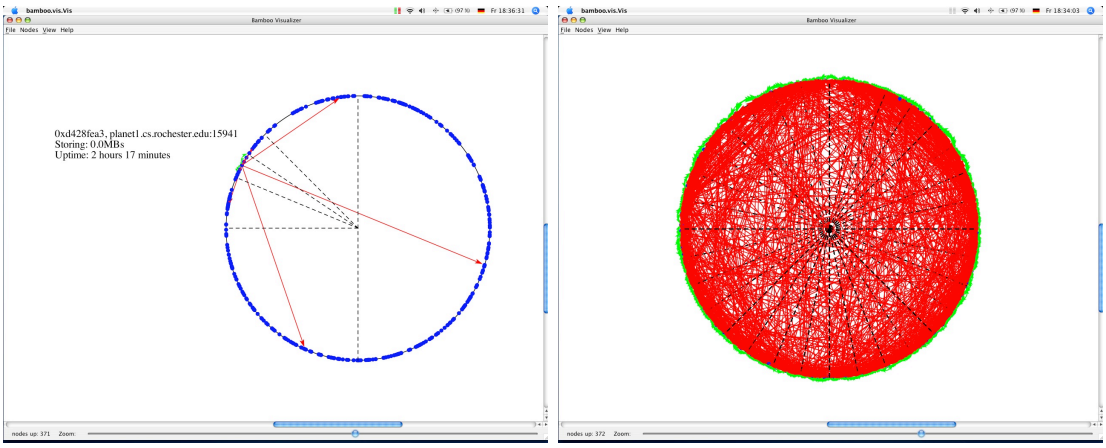
Management tools included the PlanetLab Application Manager[62] for status monitoring, *vxargs*,[63] and especially tools from the CoDeeN content distribution network project like *codeploy* for deployment of new builds, and *multiquery* for parallel execution of startup and stop commands triggering local scripts on the PL nodes.[64]

The operating systems used for the prototype include Fedora 6 on the PL nodes running Bamboo, Fedora 8, and MacOS X 10.5 for the rest of the infrastructure. The client scripts were programmed in Python, adapting and extending the short Python clients for Open DHT.[65]

5.4.4 Testing

In the following, a set of experiments using the prototype are described which have been conducted to confirm that OIDA is able to fulfill the IOT functional require-

[62] URL: `http://appmanager.berkeley.intel-research.net/` (04.2008).
[63] URL: `http://dharma.cis.upenn.edu/planetlab/vxargs/` (04.2008).
[64] Wang et al. 2004 [208]. `http://codeen.cs.princeton.edu/codeploy/` (04.2008).
[65] Rhea et al., 2005 [169]. `http://opendht.org/` (04.2008). The scripts in Appendix B extend previous joint work with my student, Ignacio Mochales Cuesta.



(a) Prototype Overlay Network (b) Prototype Routing Structure

Figure 5.7: OIDA Graphs

ments, as ONS does. In addition, some results on its performance are presented.

Document Preparation

The first step in testing the prototype involved the creation of individual address documents, for simplicity only containing fictional NAPTR records corresponding to a chosen EPC set, and no additional data fields. This is the task of the script `oida_prepare.py` (see Appendix B). The documents are encrypted by AES,[66] and an RSA signature[67] is added.[68] The results are stored locally in a Berkeley DB database instance.[69]

One major test for the record creation script involved the generation of 100,000 documents, AES-128 encryption, and RSA-2048 signature on a desktop PC.[70] The test aimed to show lower bounds for the speed of encryption and signing.[71]

This experiment took approximately 37.33 minutes, with an average speed of 44.68

[66] NIST, 2001 [141].

[67] Rivest et al., 1978 [172]; Menezes et al., 1997, pp. 433 [131]. RSA was chosen in the prototype for obtaining rough lower bounds for the signature speed; in practice, its secure application and implementation must be verified, cf. Mao, 2004, pp. 559 [127].

[68] Most cryptographic operations were implemented using the Python Cryptography Toolkit: `http://www.amk.ca/python/code/crypto.html` (05.2008). Note that for the prototype we chose to first encrypt before signing, to be able to detect possible corruption during network transmission more easily.

[69] Berkeley DB: `http://www.oracle.com/technology/products/berkeley-db/index.html` (05.2008).

[70] Pentium 4, 2.80GHz, 1 GB RAM, Fedora 8, Python 2.5 (r25:51908 GCC 4.1.2), python-crypto-2.0.1-7.1.fc7.

[71] A production implementation could use dedicated hardware and more efficient ECC, see Hankerson et al., 2004 [85].

```
Generating RSAkey ...
Done. Duration of Key Generation: 2.224539042 seconds.
100000 documents created, encrypted, and locally stored.
Total duration: 2240.18896604 seconds.
Storage duration: 2237.96417999 seconds.
Average: 0.022379642 seconds per document.
```

Figure 5.8: OIDA Document Creation

records per second (Fig. 5.8), which was confirmed in magnitude by repeated test runs. The size of the database file was 132 MB, the size of the file containing the AES-128 keys corresponding to each EPC was 5.6 MB.

We conclude that even for massive amounts of data records to be stored in OIDA,[72] the local preparation process, including encryption and signing of data, is very fast.

Document Storage

This experiment simulated the publishing of EPCIS address data by an information provider. The data had been prepared in advance by using `oida_prepare.py`, which stored the address documents in a local database. The script `oida_put.py` (Appendix B) was used to insert the documents into the DHT from a client situated in the same university LAN as the OIDA gateway. We did not use additional salts s as input for the CHF, because its impact on the performance is negligible.

For this test it was assumed that the provider uses an OIDA proxy in his own organization, which in turn contacts an OIDA node via XML-RPC to store data in the DHT. We assumed this node also to be situated somewhere near, for example in a demilitarized zone (DMZ) of the local organization, similar to externally reachable company DNS servers. This was modeled by choosing a local PL node running OIDA as a storage gateway, see the fast RTT (ping) rate between the client and the OIDA gateway in Fig. 5.9.

During the experiment, the Bamboo DHT suffered from moderate churn and network timeouts common to PL, around 2% of its nodes became unavailable – some of which reappeared later, however. The script used a timeout of 30 seconds, storing attempts taking longer than this – for example, due to network latency, load of the gateway, or storage node – were considered a failure. The number of EPCs and therefore individual documents was 2,000, each of which was stored in five copies. The average storage time per copy was 580 ms, including failed attempts and some longer durations, which raised the average in comparison to the median time of 290 ms.

[72] Cf. the later Section 5.5.1.

```
OIDA Gateway: 141.20.103.210:15942
2000 out of 2000 EPCs stored successfully (100.0%).
Statistics for all repliche:
9901 out of 10000 repliche stored successfully (99.01%).
Total duration: 5802.0598 seconds.
Median: 0.2896 seconds.
Average: 0.5802 seconds.
Minimum: 0.035 seconds.
Maximum: 30.0014 seconds.
Standard Deviation: 1.5875 seconds.
--- planetlab1.wiwi.hu-berlin.de (141.20.103.210) ping statistics ---
5350 packets transmitted, 5350 received, 0% packet loss, time 5349006ms
rtt min/avg/max/mdev = 0.189/0.331/2.154/0.087 ms
```

Figure 5.9: OIDA Document Storage

Of all the 10,000 storage attempts, about 99% were considered successful. For each EPC, at least three documents were stored successfully. This means, even in face of this loss, a client application could still resolve 100% of the EPCs to corresponding documents, which was confirmed by the retrieval experiments below. In a real application, the detailed list of failed attempts could be used for selected storage retries of more copies at later times. In conclusion, at least within the experimental settings and under moderate churn, storage to OIDA is feasible. Not surprisingly, document storage to the DHT is more than ten times slower than the document generation, but still relatively fast.

Finally, we measured the retrieval times for two different clients, representing a corporate OIDA proxy and a smart home application, respectively.

Document Retrieval from a Corporate Network

The final set of experiments measured the time to retrieve the documents stored during the tests described in the previous sections, in parallel from three different OIDA gateways around the world: Berlin,[73] Helsinki, New York (Table 5.1). The last two – arbitrarily selected from nodes with different RTTs – gateway sites served to test the feasibility of choosing remote OIDA gateways for failover, round robin, or increased confidentiality with respect to specific gateways.

It must be noted however, that the XML-RPC connections were not secured by TLS during the test, which would be necessary in OIDA. Therefore, the impact of TLS connections from clients to the OIDA gateways on the performance was not measured – however, we consider this overhead not to be critical in practice because it is possible to multiplex several application connections over the same TLS channel over a longer time, and the TLS delay is mostly dependent on this single session establishment, ideally performed once for all documents to be retrieved.

[73] The client was situated in the same university LAN as this OIDA gateway in this experiment.

The retrieval process, which is the OIDA analogon to an ONS lookup, used the script `oida_get.py` (Appendix B). Again the timeout was 30 seconds, which is reflected by the maximum and average duration of all retrieval attempts, not only successful ones. The choice of a specific timeout value is up to the client application within the limits provided by the DHT.

OIDA Gateway	Berlin	Helsinki	New York
IP Address	141.20.103.211	193.167.187.187	216.165.109.81
RTT avg. (ms)	0.32	51.40	112.24
Success EPC	100%	100%	100%
Success Replica	99.78%	99.84%	99.68%
Total Duration (s)	4924.94	7068.37	9130.31
Median (s)	0.2136	0.3870	0.5253
Average (s)	0.4925	0.7068	0.9130
Minimum (s)	0.0063	0.1536	0.3347
Maximum (s)	30.0026	30.0579	30.1139
STD (s)	1.6716	1.7883	1.9639

Table 5.1: OIDA Document Retrieval – Company

OIDA Gateway	Berlin	Helsinki	New York
IP Address	141.20.103.211	193.167.187.187	216.165.109.81
RTT avg. (ms)	11.925	42.054	103.320
Success EPC	100%	100%	100%
Success Replica	100%	99.95%	99.69%
Total Duration (s)	5139.95	6383.69	9572.74
Median (s)	0.2504	0.3621	0.5140
Average (s)	0.5140	0.6384	0.9573
Minimum (s)	0.0468	0.1328	0.3231
Maximum (s)	30.0150	30.0464	31.0519
STD (s)	1.3542	1.3191	1.9634

Table 5.2: OIDA Document Retrieval – Smart Home

Document Retrieval from a Smart Home

While in the previous test the client was situated in a very fast university network, modeling a corporate client, we also tested the retrieval of documents from a client connected via a DSL connection from Germany suffering from approximately 1% packet loss on the average during pings to the gateways. This experiment was conducted to model a possible UC application retrieving address data for EPCs, for example as would be gathered by a periodic inventory process by smart shelves.

The tests were conducted on another day, using a different EPC set of the same size, same document size, and roughly equivalent size of the DHT (330 nodes). In spite of these differences and the fluent state of PL, the results shown in Table 5.2 are surprisingly consistent with the previous retrieval experiment. Connection to remote gateways took longer and had higher miss rates due to timeouts, but were able to retrieve all documents if replication was provided.[74]

[74] During these particular experiments and similar test runs, very rarely only three copies of each document were successfully retrieved, and never less than three.

Those real-world experiments, though limited in scale, combined with the theoretical results from DHT research on scalability, give good reason to pursue further research and development on DHT-based name services for the IOT. OIDA, for example, if supported by an appropriate membership and authorization procedure and trust structure, fulfills all the functional requirements, as well as excellent scalability, and offers – as we experienced during PL deployment – appropriate robustness in face of random errors.[75]

Scalability and low latency are important non-functional requirements, which we will discuss in the next section.

5.5 Scalability and Latency

Before we discuss scalability and latency in detail, we give a rough estimate on the cardinality of the EPC space, as well as the fraction of EPCs the IOT and a corresponding name service should be able to cope with at a given time.

5.5.1 EPC Usage Estimation

In theory, what is the maximum number of EPCs that can be generated without duplicates? Figure 5.10 gives an overview of the required bit lengths for storing the different EPC classes on physical tags (last column).[76] However, the actual EPC storage requirements in general, outside of RFID tags, are depicted in the 8th column.

Taking these numbers in consideration, the IOT will in theory deal with the following maximum number E of EPCs:

$$E = 6 \cdot 2^{96} + 2^{170} + 2^{195} + 2^{198} + 2^{202} \approx 6.880 \cdot 10^{60}. \tag{5.1}$$

This does not yet take potential future extensions or other numbering schemes into account. Important for E, however, is the maximum bit length required ($l = 202$ so far), because $l + 1$ bits are currently more than enough to store all other EPC categories, too. We note that a hypothetical system capable of dealing with E could in addition also store all possible IPv6 addresses (2^{128}), in theory. This indicates the flexibility of the current EPC system to be potentially extended to IP addresses as well. However, for a practical comparison, recall that the number of all atoms on earth is approximately 10^{50}, thus very small compared to E.[77]

[75] This assumes a good replication of the data, which is very easily achieved with DHTs as our prototype shows.

[76] Image source: EPCglobal, 2007, p. 90 [51].

[77] Wikipedia, s.v. *Atom* [215] (06.2008).

Bit Field / EPC Identity Names	Reserved Memory bits	CRC-16 bits	Length bits	RFU bits	EPC/ISO Toggle bit	Reserved / AFI bits	EPC Header + Filter value bits+ Partition value bits + Domain Identifier bits	Word Boundary Filler bits	TID bits	User Memory bits	Total bits required
GID-96	64	16	5	2	1	8	96	0	32	0	224
SGTIN-96	64	16	5	2	1	8	96	0	32	0	224
SGTIN-198	64	16	5	2	1	8	198	10	32	0	336
SSCC-96	64	16	5	2	1	8	96	0	32	0	224
SGLN-96	64	16	5	2	1	8	96	0	32	0	224
SGLN-195	64	16	5	2	1	8	195	13	32	0	333
GRAI-96	64	16	5	2	1	8	96	0	32	0	224
GRAI-170	64	16	5	2	1	8	170	6	32	0	304
GIAI-96	64	16	5	2	1	8	96	0	32	0	224
GIAI-202	64	16	5	2	1	8	202	6	32	0	336

Figure 5.10: EPC Identity Types (Source: EPCglobal)

In practice, therefore, only a comparatively small fraction of E would be necessary for the IOT at any given time. In the following, we give a very rough estimate for the number of EPCs needed for items in practical use, by presenting three different scenarios.[78]

Scenario 1 (Small Scale Adoption)

In all of the following, we focus on SGTIN EPCs, probably the most important EPC identifier class for a future IOT. In the first scenario, let us assume the number of companies participating in the IOT at a time in near future is 10,000 ($c = 10^4$), further, that the *maximum* number of items produced per company per year is one billion ($i_{max} = 10^9$), and the *average* number of items produced per company per year is one million ($i_{avg} = 10^6$). Further assume that item creation and discarding rates are equal, and that on average at a given time the item production of the last $t = 2$ years stays in use.

Given these numbers, the average number $E_{s.avg}$ of EPCs used in practice will be approximately:

$$E_{s.avg} \approx c \cdot i_{avg} \cdot t = 2 \cdot 10^{10}. \tag{5.2}$$

We also assume a maximum number of object classes per company $o_{s.max} = 10^6$, and a corresponding average $o_{s.avg} = 10^3$, already reflected in the numbers of EPCs

[78] Again it should be stressed that these scenarios are attempts to roughly forecast the future adoption of the EPC and IOT, and may be inaccurate. Further research in that direction should prepare the ground for more accurate estimates.

above.

Scenario 2 (Medium Scale Adoption)

Here we assume the number of IOT companies is 100,000 ($c = 10^5$), the maximum number of items produced per company per year is again one billion ($i_{max} = 10^9$), and the *average* number of items produced per company per year is ten million ($i_{avg} = 10^7$). Further assume that items stay in use for $t = 3$ years.

$$E_{m.avg} \approx 3 \cdot 10^{12}. \tag{5.3}$$

In this scenario we again assume a maximum number of object classes per company $o_{m.max} = 10^6$, but an average of $o_{m.avg} = 10^4$.

Scenario 3 (Large Scale Adoption)

Finally, let us assume a future IOT with one million companies ($c = 10^6$),[79] $i_{max} = 10^{10}$, $i_{avg} = 10^8$), and $t = 5$:

$$E_{l.avg} \approx 5 \cdot 10^{14}. \tag{5.4}$$

In this large scenario, let the maximum number of object classes per company be $o_{l.max} = 10^7$, and the average be $o_{l.avg} = 10^5$.

So even in the large scale scenario 3, the fraction of EPCs in use will be much smaller than the theoretical maximum. This result has beneficial implications for IOT scalability as we will see next, but also negatively affects query confidentiality in OIDA if salts are not used, as will be discussed in Section 5.6.5.

5.5.2 Class-Level vs. Serial-Level Resolution

In the discussion on scalability and storage requirements for OIDA and ONS, the following parameters will be used: N = number of OIDA nodes (analogously, number of ONS leaf servers,[80] or IOTNS nodes in general); g = average number of gigabytes (GB) of pure data storage available per IOTNS node; d = average size of a replica document in GB; r = average redundancy parameter, i.e., average number of copies for any EPC document.

[79] There are about one million registered Company Prefixes today, according to GS1: `http://www.gs1.org/productssolutions/barcodes/implementation/` (09/2007), cf. Section 4.2.2. Not all registered companies will participate in the IOT.

[80] Leaf servers are those ONS servers that actually store NAPTR RRs and form the leaves of the tree, and do not only provide glue records for delegation, as the ONS Root servers do.

We will now discuss some examples of storage demands and their technical feasibility, considering the three scenarios for IOT adoption in the previous Section 5.5.1.

Small Scale Adoption

In this scenario, the following should hold for the storage capacity C of the IOT name service, if serial-level lookup is used: $C \geq E_{s.avg} \cdot d \cdot r$. For average document size of 1 KB ($d \approx 10^{-6}$), and replication parameter $r = 6$: $C \geq 1.2 \cdot 10^5$ GB. This leads to an average storage need per company of $C_{i.avg} \geq \frac{C}{10^4} = 12$ GB, and a maximum $C_{i.max} \geq i_{max} \cdot t \cdot d \cdot r = 10^9 \cdot 2 \cdot 10^{-6} \cdot 6 = 1.2 \cdot 10^4$ GB ≈ 12 TB for the company with the highest production. Thus on average, assuming storage capacity $g = 2$ TB per node, in theory only every 166th company would have to deploy a node to fulfill the IOTNS storage requirements.

Medium Scale Adoption

$C \geq E_{m.avg} \cdot d \cdot r = 1.8 \cdot 10^7$ GB. This results in an average storage need per company of $C_{i.avg} \geq \frac{C}{10^5} = 180$ GB, and a maximum of $C_{i.max} \geq i_{max} \cdot t \cdot d \cdot r \approx 18$ TB for the company with the highest production. Given that this level of adoption will happen in future, it can be reasonably assumed that $g \geq 2$ TB. Therefore, even the company with the highest production can fulfill the storage requirements by providing less than nine servers in this scenario.

However, for those companies with a large production, the storage procedure of documents to the IOTNS will have to be optimized if serial-level lookup is to be offered, e.g., by massively parallel storage, perhaps also from several sites. Assuming the storage procedure to OIDA could be optimized to taking only 0.1 seconds on average for the transfer, and assume 10 company sites performing 100 storage operations each (to different gateways) in parallel, the time of storage would take $10^9 \cdot 10^{-1} \cdot 10^{-3} = 10^5$ seconds ≈ 28 hours to transfer all documents, without replica. If current or future DHT implementations are able to cope with this load is an important issue for future research. On the positive side, only for bootstrapping the IOTNS the whole record set of a company would have to be transferred to the DHT. If TTL values of records and salts in OIDA are chosen longer than the average value in DNS,[81] the average update rate could be much better manageable.

In general, this problem of scalability for serial-level lookups does not only affect P2P-ONS like OIDA, but also ONS, e.g., for the replication of the master ONS server of a large-production company to its slaves residing in different networks for enhanced availability. Similar to DHTs, it is unclear if DNS software could handle these massive data flows, and how long this would take.

[81] Often only one day, according to Ramasubramanian and Sirer, 2004, [162].

In summary, concerning serial-level IOTNS lookups, already the medium scale adoption scenario borders on the technically feasible today and in the near future, due to the massive amount of data necessary for supporting the companies producing the most objects. In contrast, the average production, or class-level lookups seem to pose no critical technical challenges.

Large Scale Adoption

$C \geq E_{l.avg} \cdot d \cdot r = 5 \cdot 10^{14} \cdot 10^{-6} \cdot 6 = 3 \cdot 10^{9}$ GB. Then, the average storage need per company is $C_{i.avg} \geq \frac{C}{10^6} = 3000$ GB ≈ 3 TB, and a maximum of $C_{i.max} \geq i_{max} \cdot t \cdot d \cdot r = 10^{10} \cdot 5 \cdot 10^{-6} \cdot 6 = 3 \cdot 10^{5}$ GB $= 300$ TB for the company with the highest production.[82] Therefore, for this large scale adoption scenario, it seems possible to cope with the average storage and transfer needs for serial-level lookup, but the companies with the largest productions pose challenges quite beyond what appears practically feasible in near future. Class-level lookups, however, would still be possible, even with today's technology.

As a conclusion to this section on scalability, we can state the following. Even if the IOT is adopted on a large scale comparable to Scenario 3, the class-level and average serial-level lookup performance requirements seem to be satisfiable by ONS and OIDA in future. However, starting from a medium diffusion, the companies with the largest productions should refrain from offering serial-level address information for *all* of their items, for example by selecting object classes whose serial ranges will not be published individually to an IOTNS, only as a class-level address.

To support this technically, a lookup convention could be introduced to first search for class-level information for every EPC (e.g., by setting the serial part to zero before applying the CHF). The retrieved document could carry a flag indicating that for this object class serial-level lookup is available.

Having discussed scalability with respect to number of companies and EPCs, we now turn to further performance issues, notably the problem of update propagation and lookup latency.

5.5.3 Update Propagation and Lookup Latency

Update propagation is the process of distributing new versions of data throughout a distributed system. With IOTNS, if the address list for a given object identifier changes, all documents should be updated to reflect this. If a name service depends heavily on passive caching like DNS, update propagation depends on the TTL value of the data records. In comparison, DHTs offer faster update propagation, because the positions of all document copies are easily determined by the replication pro-

[82] This does not even take EPCIS database storage requirements into account.

cedure. If a proactive caching layer is used on top of the DHT (see below), it can actively redistribute the new versions to the other storage nodes, without waiting for the passing of a TTL value.

Another important challenge in designing a name service is the problem of lookup latency or delay; i.e., the time it takes between the issuing of the name query and the return of an answer to the client. With applications designed for human interaction, low latency is critical because waiting times of more than a few seconds seem unacceptable for a human user, e.g., while surfing the Web. But other applications like email can tolerate longer latency, and higher time-out values could be set before a lookup attempt should be considered a failure.

Today it is not clear yet, if very low latency, e.g., below two seconds, would be critical for IOT applications, or only generally preferable for system performance. The EU BRIDGE Project's discovery service requirements document states the duration of a few seconds to be acceptable for EPCIS Discovery Service operations,[83] which may also be applicable for other IOTNS. However, this datum was extracted from a questionnaire with a low count of responses, only.[84] We note that in our smaller-scale experiment of a few hundred nodes on PL described in Section 5.4, the median and average lookup latency is well below one second.

Regarding latency, DNS has the advantage of caching not only direct results on the lookup path, but also addresses of servers higher in hierarchy that can also be used for similar queries, e.g., for different names belonging to the same domain. This could often render DNS faster in practical use than DHTs in large-scale systems, even though the latter are also very fast, needing only $O(\log(N))$ hops to resolve the query.[85] This was already noted in early proposals for using DHT for DNS.[86] The caching advantage of DNS applies only to the more popular domains, not to the long tail of the query distribution, which was empirically shown to follow a Zipf distribution; i.e., a power-law,[87] where the number of requests for the k-th most popular record is proportional to $k^{-\alpha}$, with $\alpha \approx 0.91$ for DNS.[88] There are also results uncovering further client-side DNS problems, which also affect DNS latency negatively.[89]

Reactive or passive caching in DHT, which means storing query results on the lookup paths, seems to have no notably positive effect on DHT latency. In con-

[83] BRIDGE, 2007, p. 9 [22].

[84] A total of 15 companies responded according to BRIDGE, 2007, p. 8 [22], out of which only five responded to relevant items Q81 and Q82, p. 50.

[85] Latency is often quantified as the average path length in a network of size N, a simplification assuming that hop by hop latency is roughly constant and independent of lower network layers.

[86] Cox et al., 2002 [37].

[87] For the astonishing ubiquity of the power-law distribution in natural, social, and technical systems cf. Newman, 2005 [139].

[88] Jung et al., 2001 [100]; Jung et al., 2002 [101]. However, research on DNS queries and performance in general faces methodical difficulties, especially the problem of how representative the analyzed traces are, see Liston et al., 2002 [121]; Pang et al., 2004 [148].

[89] Park et al., 2004 [151].

trast, however, an additional proactive caching layer could be used, which actively replicates the most popular items according to a distributed estimation of the query distribution, as was successfully implemented and tested for Zipf query distributions with BeeHive in the CoDoNS project, which can offer constant lookup performance with moderate network and storage overhead.[90]

Similar self-adapting caching layers could be applied to OIDA or other P2P-ONS systems, and could be optimized once reliable data on the query distribution of object identifiers or object classes becomes available. The impact of those extensions on other metrics, especially on security requirements of different stakeholders, is an important topic for future research. As a preliminary remark we note that data availability, for example, could be positively affected, but query confidentiality possibly in a negative way due to the automatic profiling of client behavior, even though the keys queried for are hashed and the documents encrypted. However, in BeeHive this frequency analysis is conducted in a distributed fashion without sending the results to a centralized server, and equals in scope analysis potentials that already exist on DHT nodes.

To tie in with this topic, we now turn our focus to security aspects of OIDA and P2P-ONS in general.

5.6 OIDA Security

5.6.1 Overview

In this section, we discuss how well OIDA does fulfill the initially stated security requirements in Section 2.3, what assumptions on the environment are important for OIDA security, and what technical and organizational security measures should flank its deployment and use. We mainly – but not exclusively – focus on security aspects of the underlying DHT, which are new in comparison to the rather classical security aspects of the clients and the PKI. For iterations of the security analysis process in future stages of implementation, certainly all OIDA components and their interplay will need to be reanalyzed in-depth.

The main adversary model in the literature is that of an insider of the DHT system, controlling one or multiple nodes (*malicious nodes*).[91] This is in line with the frequent assumption in P2P research that system membership is open and free for every interested entity. In contrast, we assume P2P-ONS systems like OIDA to be *infrastructure networks* with controlled membership, regulated by business contracts. This can be compared to ONS, where a central institution (EPCglobal) issues EPC ranges and grants membership only in exchange for payment.

We do not propose the same central business model of a single entity controlling

[90] Ramasubramanian and Sirer, 2004 [161]; [162].
[91] Sit and Morris, 2002 [187]

membership to OIDA, but do assume the existence of membership and authorization procedures combined with strong incentives for members to support the system functionality, reflected and flanked by a technical trust infrastructure, for example a web of trust or PKI. In addition, those business contracts should – besides functional and performance requirements – also try to enforce system and data security.

Under those assumptions, insider attacks on availability and integrity of the system and the data it provides become less probable than in a completely open environment because of the self-interest in the system, and the detection risk combined with possible legal and business retaliation.[92] Confidentiality requirements, however, can often be violated by insiders without a major risk of detection, and should be inherently enforced by the system.

Not least, as they are common in today's Internet, there are also external adversaries who need to be considered, for example ISPs controlling routers for data collection,[93] or criminals threatening to perform DoS attacks.

The following sections discuss availability and multipolarity, integrity, as well as confidentiality and anonymity.

5.6.2 Robustness and Availability

OIDA should be available in face of random errors – this corresponds to the robustness requirement in Chapter 2, which will be mostly subsumed in the discussion here – as well as in face of targeted (D)DoS attacks.[94] There are at least two further aspects of availability: availability of the name service as a system, and of the actual data that is stored.

System Availability

First, DHTs offer no single point of failure, a major advantage over the DNS, where the root and TLDs are attractive potential and actual targets for DoS attacks (see Section 3.4.1). For adversaries aiming to disrupt the service of a specific company, no single and clearly recognizable target is presented.

Furthermore, the nearest neighbors of a node in the overlay topology, for example those in a Bamboo leaf set, are highly interconnected among each other, which offers a robust way of sending messages in case of long range failures, if nodes in the routing table are currently unavailable. This robustness applies to directed attacks and random errors. There are several studies that have confirmed DHT robustness,

[92] It is possible, however, that under specific circumstances members would conduct such attacks against other members, therefore corresponding countermeasures combined with monitoring should be available.

[93] For traffic analysis at the AS-level cf. Murdoch and Zielinski, 2007 [137].

[94] For general network countermeasures against (D)DoS cf. Peng et al., 2007 [152].

we discuss only one partially contrary opinion here. In one comparison of DNS with DHTs, the DHT robustness against attacks was shown to be high, but with respect to random errors it appeared to be lower than in DNS.[95] But in this study the effect of *sequential neighbors* was explicitly omitted, which constitute a fundamental element of robustness against random errors.

A famous attack strategy in P2P – and ad hoc – networks is often called the *Sybil Attack*:[96] An adversary generates and adopts many IDs to control a large part of the overlay network. This can be used for DoS against routing, or for attacks against data and message confidentiality and anonymity, because the adversary will be able to analyze much more network traffic than normal. Related is the so-called *Eclipse Attack*,[97] during which an adversary controls so many nodes and overlay paths that he is able to completely suppress or otherwise control the traffic to a target.

Both attacks – and similar attacks on routing – can be mitigated in OIDA by using a Node PKI for member nodes, and by cryptographically signing overlay maintenance and routing messages. This is possible due to our assumption that OIDA will be deployed as an infrastructure network with controlled membership, in contrast to arbitrary P2P systems. We discuss such a Node PKI briefly in the following.

Node PKI. Because OIDA is an infrastructure network with defined membership procedures, node secret keys, or, for better scalability, public / private key pairs in conjunction with certificates signed by the CA can be used for fall-back secure routing.[98] This system to authenticate will be called *Node PKI* in the following.[99]

This Node PKI can be used as well as to confirm the non-existence of records. In addition, a PKI would be highly useful for the record publishing phase to authenticate the OIDA node against the publisher, when publisher certificates could also be used for mutual authentication (see Section 5.6.4). Node public keys could also be used as an input to the CHF for generating overlay node IDs, providing a strong link between overlay ID and public key. Those IDs are in general known as cryptographically generated identities, and are commonly used in ad hoc networks without available CA.[100]

The certificates issued by the CA – or a web of trust – including signed member public-keys, and flanked by organizational policies which would deny entities from using unnecessary many overlay addresses, could prevent a single party from adopting large sets of overlay IDs.

[95] Pappas et al., 2006 [150].

[96] Douceur, 2002, [47]; Danezis et al., 2005 [40].

[97] Singh et al. 2004 [186].

[98] For a corresponding extension of Pastry, see Castro et al., 2002 [27].

[99] However, every PKI based on a central CA can itself create new security risks, cf. for example Burmester and Desmedt, 2004 [25]. Alternatives are trust-graphs that do not have only one single root, like a web of trust.

[100] Blaß et al., 2007 [17]. If no CA is available, cryptographic puzzles could be used to slow down Sybil Attacks, cf. Baumgart, 2008 [13].

Public-key cryptography on the nodes, however, will generate more overhead than standard routing, a situation comparable to DNSSEC (see Section 4.3.1). But given that the signatures on the stored data can provide end-to-end authenticity (from publisher to client), the Node PKI is not necessary for verifying the authenticity of retrieved data, and could potentially be confined to special circumstances, like node joining, data publishing, and the verification of a non-successful query.

Data Availability

To make the data stored in an IOTNS robust against random failures and denial-of-service attacks, multiple copies of each document d should be stored, preferably at different physical locations.

With OIDA, this can easily be achieved by the individual publisher without the administrative overhead that is necessary for distributing and maintaining DNS servers over multiple regions in the real world. During data storage, nearly arbitrary redundancy can be achieved by the choice of a redundancy parameter r_{max} indicating the number of copies stored to the underlying DHT. This needs to be supported by a public convention among publishers and clients that defines how the replica pre-images are constructed before the CHF $h(x)$ is applied.[101] Let $h(a, b, c)$ denote the value of the CHF h for the concatenation of a, b, and c.

Example 1. *Agreement 1 for generating overlay IDs I_j of r_{max} copies of the document corresponding to the same EPC e, using a fixed salt s:*

$$I_1 = h(e, s, 1), ..., I_{r_{max}} = h(e, s, r_{max}). \tag{5.5}$$

Another agreement could be to apply different CHFs for different copies, or the same CHF multiple times.

Example 2. *Agreement 2 for generating overlay IDs I_j of r_{max} copies of the document corresponding to the same EPC e, using a fixed salt s:*

$$I_1 = h(e, s), I_2 = h(h(e, s)), ..., I_{r_{max}} = h^{r_{max}}(e, s). \tag{5.6}$$

Note, however, that Agreement 2 would allow linking analysis by adversaries who know only one intermediate hash value $h^j(e, s)$, which would let them calculate other replica locations simply by applying the CHF again without knowing the pre-image. This could lead to reduced client confidentiality or anonymity in the long term because now the nodes to be observed for specific queries are known, at least until the salt is changed. DoS attacks could also focus on all the nodes that store replica of this particular document. This, however, would have to be conducted

[101] There is an intrinsic trade-off between the number of copies, and the available storage space and bandwidth, especially for serial-level lookup, see Section 5.5.1.

rather blindly by an adversary, since he cannot link those documents to a specific EPC.

Of more practical value to an adversary would be focused attacks against all the nodes carrying a document for a known EPC, simply by applying the same public convention as needed for storage and retrieval of replica documents. For this to work, however, the salt s needs to be known as well. Another advantage of DHT over DNS is that due to the nearly uniform distribution, no conventional adversary could attack all nodes that store documents for a particular larger company – this could be comparable to an effort to attack nearly all DHT nodes at the same time.

In addition to publisher-controllable replication, many DHTs offer automatic replication options implemented by their storage layers. Bamboo, for example, has a `min_replica_count` parameter in its node configuration file (see Appendix A), which could be combined with proactive caching layers for the DHT such as BeeHive to reduce lookup latency.[102]

5.6.3 Multipolarity

Multipolarity – in our technical definition as resistance to *Blocking Attacks* conducted by countries – is a special instance of the availability requirement with respect to countries as counter-stakeholders (see Section 4.2). Compared to ONS or MONS, how multipolar is OIDA? First let us reconsider the blocking attack in the context of OIDA.[103]

Definition 4. OIDA Blocking Attack. *The Blocking Attack in OIDA occurs if one Country A blocks access from Country B to all of the OIDA nodes situated in A.*

First we note, that a blocking attack in OIDA would be only practical at the border router level, otherwise all companies within A would have to install corresponding firewall rules or application-level filters whose distribution seems impractical. The lack of a central root compared to ONS makes such an attack therefore difficult to conduct. The risk of probable retaliation – that is, B blocks all access from A to all nodes in B – is also difficult to calculate for A: the OIDA nodes in B carry a nearly unpredictable assortment of documents, many of which could be critical for companies in A, especially also documents published by companies from A to OIDA.

Therefore, for rational nations, a blocking attack in OIDA would in general hardly look promising. But what about data loss for B, if such a case occurs? In the following we show that OIDA data replication can also avoid – with high probability – the situation that all copies of a document would be stored within just one country; potentially the blocking one. This would at least guarantee business continuity for trading of national goods in B, as well as international access to address documents

[102] Ramasubramanian and Sirer, 2004 [161]. Cf. Section 5.5.3.
[103] Compare the definition in Section 4.2.1.

for every EPCIS located in B. Assume A to be the country that controls the largest number n_A of a total of N OIDA nodes. Then the fraction $f_A = \frac{n_A}{N} \in [0, 1]$. For a given EPC e and its corresponding document d, let p be the probability that a single copy of d is stored at an OIDA node of country A. Let *ideal OIDA* be based on a DHT with a large country and node membership, using an ideal CHF with nearly uniform output probability distribution.

Proposition 1. *For ideal OIDA, $p \approx f_A$.*

Proof. Ideal OIDA has a nearly uniform distribution of document identifiers and nodes across the identifier space. □

Let us assume that replication is initiated by the publisher during application of the CHF, for example by the method of generating replica IDs presented in Example 1.

Proposition 2. *For ideal OIDA, the probability that all k copies of a given document d are stored in Country A is approximately p^k.*

Proof. The uniform choice of storing nodes at every replication step is independent from the choice during the other steps. □

Lemma 1. *The probability that at least one copy of a given data document d is stored outside of A, is approximately $1 - p^k$.*

Therefore, the number of copies can be chosen in such a way that a nearly arbitrary low blocking risk even with respect to the most powerful Country A can be achieved.

Example 3. *Consider the extreme case of a single Country A controlling half of all OIDA nodes. Let the replica count be $k = 10$. Then the chance that at least one copy is stored outside of A is $1 - 0.5^{10} > 0.999$.*

In conclusion, in OIDA the risks involved with blocking attacks would be nearly incalculable for the attacker. In addition, documents can be replicated in such a way that with high probability not all copies are stored in a single country. Additional replication during such an attack would be easily feasible.

5.6.4 Integrity

OIDA aims to provide end-to-end security and version control implemented within the stored documents. The DHT nodes should in general not necessarily be more trusted than ONS servers with respect to document integrity and authenticity. Data authenticity can be established by letting the publishers sign the data to be stored, and by using an external trust and certificate infrastructure – possibly a hierarchy

with a distributed root like in Multipolar ONSSEC (see Section 4.3.3), or a web of trust.[104]

This offers flexibility for changing the underlying DHT layer if necessary, and causes in general operations no additional delay and overhead for document handling on the actual DHT nodes, besides the possible use of a Node PKI to secure DHT membership and routing (see Section 5.6.2).

Spam Protection. Another issue would be the avoidance of unsolicited data entries in OIDA. Normally those would be easy to filter out by a client due to the lack of a genuine signature. However, they could slow down the performance of the whole system. Therefore, the verification of a publisher's certificate – and possession of the corresponding private key – by the OIDA node used for publishing would be necessary. This could be implemented on a per connection base, offering good performance, or even on a per document base, if internal members should be originators of spam.

5.6.5 Confidentiality

In this section, we discuss OIDA's ability to satisfy the confidentiality requirements identified in Section 2.3.3. First we discuss an important prerequisite, the feasibility of key distribution.

Key Distribution

A central question for estimating the kind and strength of the cryptography that can be used to achieve confidentiality goals is: Will there be a global PKI in place that makes the use of public key cryptography and certificates possible, especially on the client side? Will there be something more than the EPC that is shared between information provider and client? Will they share a common parameter, perhaps even a secret?

It is in our opinion probable that for supply chain use of the IOT a global PKI could be established, for example as part of the security services announced by EPCglobal.[105] If such a PKI, perhaps in form of a web of trust, could scale to arbitrary information publishers and clients in an extended IOT or to Ubiquitous Computing must be considered an open problem.

Related is a similar problem: the distribution of shared secrets for RFID tag access control or deactivation procedures (*kill passwords*), which are already part of RFID standards,[106] or are about to be included due to security requirements concerning

[104] Burmester and Desmedt, 2004 [25]; Zimmermann, 1995, [218].
[105] X.509-based PKI for the EPCglobal Network is indicated by EPCglobal, 2008 [54].
[106] EPCglobal, 2007 [52].

the tag and reader communication. If the latter could be solved, perhaps even for open UC environments, the distribution of keys to OIDA clients could be established as well.

Depending on the state of key distribution that can be assumed, different modifications to OIDA can be designed, which offer different degrees of confidentiality. As indicated earlier, many security features can be implemented at the document level. For data authentication (e.g., similar to DNSSEC), this was already shown to be straightforward by signing the information before storing it.

In the following, we discuss the options for document access control in OIDA to fulfill a publisher's (potential) confidentiality requirements for the address or object data, and for satisfying client confidentiality requirements.

Provider Confidentiality Requirements and Access Control

If access control on the stored data is required by the publisher, he needs to offer a way for clients to authenticate themselves, e.g., by issuing shared secrets or using public-key cryptography. Those keys need to be distributed using secure channels, in general separately from the actual system in use.

The same key material, however, could be used by the information provider P to encrypt the document d stored in the DHT. If necessary, multiple copies encrypted by different keys can be stored in the DHT, or different information documents for different clients – both approaches, however, could increase the storage space and time needed. Another solution would be to store common documents for the same user group sharing the same key.[107]

To locate these documents, the cryptographic hash value could be computed using the EPC and the key together as a pre-image, basically resulting in a message authentication code (MAC).[108] Even though third parties could analyze the network traffic or locate the document on their own, they could not read it without the corresponding shared or private key. Against those adversaries, client confidentiality would also be enhanced at the same time.

Client Confidentiality

Client confidentiality – in the context of using an IOTNS – mainly depends on the following data: query source IP, query content equivalent to a (partial) EPC, returned document, which may contain information equivalent to the EPC, or parts of it. With ONS, the final ONS server address could be equivalent to the EPC

[107] In general a bad security practice, but perhaps acceptable for transferring address data or a single EPC.

[108] Menezes et al., 1997, pp. 352 [131].

Manager field (see Table 2.2). With OIDA, because of the apparently random assignment of documents to nodes, this does not apply.[109]

The confidentiality of the query source IP is also part of the anonymity requirement that we will discuss later. In the following, we will investigate what security measures would be applicable for hiding the query and reply content from third parties, including the OIDA storage node.

Strong Confidentiality Scenario. Though it is likely that some kind of global PKI will be run by EPCglobal for supply chain use, it is not clear yet if this would be opened for or even scale to the much larger set of possible Ubiquitous Computing applications. Therefore, assuming a global PKI to include all UC clients seems to be a very strong requirement.[110]

On the other hand, if this is restricted to supply chain networks, or even to clients of a particular information publisher only, such a PKI may become possible. It may also be used to securely exchange shared secrets between provider and client, or lists of lookup salts for an additional input to the CHF (see below).

We will refer to a setting involving PKI as a *Strong Confidentiality Scenario*, with other counter-stakeholders than the publisher in mind. The publisher must have at least the information that a particular client *might* retrieve the document once, but he will not be able to observe the actual access, an improvement to ONS. If OIDA and EPCIS access are kept separate, more detailed information on the client and its interests can be gathered by the provider if he also controls the EPCIS. This could be only prevented by using stronger measures to enforce client confidentiality like onion routing or PIR (see Sections 4.4.3 and 4.4.4), which would pose a possible conflict with access control measures to enforce a publisher's confidentiality requirements.[111]

In this scenario, which also matches the ideal situation for provider confidentiality discussed in the previous Section 5.6.5, the following confidentiality requirements are satisfied, cf. Table 5.3.[112]

Medium Confidentiality Scenario. Furthermore, if there is no PKI, can we assume the existence of shared keys between information providers and clients? For EPC tags, kill and access passwords must be transferred securely from manufacturer

[109] Though some information is leaked by the node address, which might be used for long term analysis.

[110] Lopez et al., 2005 [125].

[111] A future research topic would be to investigate the ability of protocols like Direct Anonymous Attestation used in *Trusted Computing* to solve this conflict, cf. Brickell et al., 2004 [21].

[112] Local gateway refers to the local OIDA gateway in all of the following, which is either part of the company, or will be contacted via TLS, so that outgoing queries from the gateway are part of the anonymity set of queries routed through the local gateway. Long term analysis by a local ISP observing this gateway might break anonymity and location confidentiality.

Confidentiality Requirement	**OIDA**
Shared Confidentiality Requirements: Provider and Client	
Address-Data Confidential?	✓
Object-Data Confidential?	✓
Confidentiality Requirements: Provider	
Identity Confidential?	no
Client Confidentiality Requirements	
Counter-stakeholder: OIDA Node, ISP, Internet Backbone	
Query Time Confidential?	no
EPC Manager Confidential?	✓
EPC Object Class Confidential?	✓
EPC Serial (if used) Confidential?	✓
EPCIS Document Confidential?	✓
Source IP Confidential?	✓(not w.r.t. local gateway)
Anonymity?	✓(not w.r.t. local gateway)
Query Confidentiality?	✓
All Trackable Identifiers Confidential?	no: $h(e, s, r)$
Location Confidential?	✓(not w.r.t. local gateway)
Unobservability?	no

Table 5.3: Strong Confidentiality Scenario

to the point of sale and finally to the end user – it would be easy to transfer another password k on the same channel for accessing online information. k could then be used to encrypt the document d before storage, and to decrypt it after retrieval.

Often, though, shared secrets do not scale well, are hard to manage and distribute securely, and have huge usability problems if there is no management device (e.g., a PDA) at hand, which itself could become a target for attacks. However, secure key distribution does seem in general very difficult in practice, but not impossible. Some recent research in this direction[113] for example focuses on splitting the key material – or even the EPC itself – and on sharing it only successively in time, or even distributing it across several RFID tags, possibly using threshold cryptography (see Section 4.3.3).

How important key distribution will turn out for OIDA will be investigated in the next scenario where we assume no pre-established key material for securing the lookup process.

Low Confidentiality Scenario. What can be done, if information provider and client share nothing but the EPC? The CHF value is in theory computed over a pre-image space of at least 2^{88} possible inputs – the space of all possible SGTIN-96 EPCs, disregarding the fixed header bits. This would not be bad as a protection even against more advanced attackers. In practice, however, only a small fraction of this space would be in use at a given time (see Section 5.5.1).

Depending on the development of RFID, it is quite probable that the necessary number of EPCs to precompute the hash values in a *Dictionary Attack* is small in comparison, e.g., possibly as low as $2 \cdot 10^{10} < 2^{35}$ in a small adoption scenario for

[113] Langheinrich and Marti, 2007 [117]; Juels et al., 2008 [99].

the IOT. A corresponding dictionary would take less than 1 TB of storage space. In addition, the EPC is highly structured (see Fig. 1.1), and serial numbers might be created in a regular, non-random fashion. This would further reduce the effort to derive the pre-image EPC from a captured hash value.[114]

The confidentiality of the EPC and – therefore of its parts – could not be guaranteed in this scenario. It would, however, require a little more effort to infer the EPC from the hash than reading it in plain text ONS records or traffic – even more so in medium to large EPC and IOT adoption scenarios with a larger range of possible pre-images.

Salts. A potential middle course between low and medium confidentiality scenarios would be to find a way to share a salt s between provider and client, a randomly generated number of sufficient length, for example 128 bit.[115] Besides EPC e and replica number r, s would also be used as shared input to the CHF to generate the overlay ID: $h(s, e, r)$. For a third party not in the possession of s, it would be infeasible – with respect to time and storage space – to generate a corresponding dictionary that maps hash values to pre-images.

The salt could also be distributed with the tags, or like the EPC be directly stored on the tags, and possibly be protected by currently emerging tag access control measures. Each authorized party in control of the tag could read the EPC and salt from the tag, and query OIDA for corresponding EPCIS address documents.

If the key-distribution problem is not solved, the EPC itself might be used as a key for insecurely and superficially encrypting the data. The salt, assuming it is randomly generated, could fulfill the function of a key much better, at least for pure address data with lower confidentiality requirements, but would then have to be treated as a shared secret during its distribution; if the group of stakeholders acquiring the tag and salt during their distribution through the supply chain matches the group of authorized OIDA clients for the corresponding address document, and those stakeholders trust each other with respect to the confidentiality of the queries they issue for this particular EPC, this might be viable. Search spaces for attacks on the encryption key would be equivalent to those for the hash dictionary attack, e.g., relatively small in case the EPC is used as a key in a low adoption scenario.

As an additional organizational procedure, it should be investigated whether the data in the returned document could be modified to contain as few information about the EPC in question as possible, for example offering no more information in the URL than is equivalent to the EPC Manager, in order to reduce inference possibilities. This would need to be supported by EPCIS infrastructure operations on the provider side, for example avoiding externally visible directory structures

[114] There is a time-memory trade-off involved in this pre-computation attack, the storage overhead can be reduced by so-called *Rainbow Tables*, cf. Oechslin, 2003 [144].

[115] For using salts, cf. Morris and Thompson, 1979 [134]; RFC 2898, Kaliski, 2000 [103]. Salts are for example used to secure UNIX or Apache password hashes.

which may betray further parts of the EPC.[116]

Class-Level Lookup. For any serial-level document d, not only the EPCIS address for the specific item at hand can be included in d, but more options may be added to reduce query overhead and increase flexibility. *Class-level* addresses would hint at EPCIS servers for the whole Object Class. Those class-level EPCIS addresses can be cached, and would be contacted for future lookup of items of the same object class. Furthermore, d could contain a *class-level salt* s_{oc} in combination with a salt expiration date t.

This salt s_{oc} can then be used to retrieve the address document d_{oc} of an EPCIS for the Object Class (OC) of the EPC that was queried for – by serving as an additional input to the CHF besides the partial EPC. This would allow for flexibility of changing the OC-level address by changing just one document in OIDA: d_{oc} – aside from repliche for this entry. Every new version of serial-level documents could update that salt s_{oc} when t has passed.

Using this method, the creation of dictionaries for profiling class-level lookups would be harder, allowing for a limited time-frame for construction only.[117]

Serial-Salt Distribution and Updating via OIDA. In a similar way to class-level salts, serial-level salts might also be distributed in OIDA, possibly offering another option apart from letting them accompany the tags. The OC-level documents could include a list of salts usable for a specific time frame, each corresponding to a serial number range.

There are two cases. In the first case, a client has resolved an EPC e_1 of the same OC before by retrieving d_1, and t has not passed. Then, to resolve an EPC e_2 of the same OC, the client first queries for the OC-level document d_{oc} using the salt s_{oc} from d_1. To make casual analysis harder, d_{oc} should be encrypted using s_{oc} as a key. If a class-level EPCIS exists, this could be queried now by the client, sending the specific EPC via TLS. If such an EPCIS does not exist, a salt s_2 corresponding to the serial range of e_2 is retrieved from d_{oc}. e_2 is then resolved by using s_2. A more than casual adversary could break this protocol by resolving an EPC of the same OC by using a valid salt. However, which OC is to be used must be determined by some means, for example by observing a later EPCIS connection by the client.

In the second case, no OC-level salt is known in advance. For this case, there could be a copy of d_{oc} stored at the CHF value resulting from the partial EPC alone without

[116] Avoiding for example a directory name generation convention depicted in the last NAPTR record of `oida_prepare.py` (Appendix B): `http://www.example.com/prdct/ + str(epc) + /info`, and using the cryptographic hash of the EPC instead.

[117] In addition, there will be flexibility for coping with future speed-up of rainbow table construction by narrowing the time frames specified by t, at the cost of possible overhead if the time interval indicated by t becomes smaller when compared to the record TTL of the actual serial level documents.

any salt as additional input. This involves another potentially severe tradeoff for the client's confidentiality. If this OC-level query is captured, the EPC Manager could be determined by a dictionary attack. Even worse, an adversary could also retrieve d_{oc}, and, due to the lack of good encryption of d_{oc} in this case, could with little effort calculate all possible pairs $((e, s), (h(e, s))$ to identify further EPCs of the same OC if he is able to capture the corresponding network traffic. Therefore, no class-level documents that are not protected by salts should be stored in OIDA if they contain salt lists for serial lookup, leaving the question of bootstrapping in the second case open.

In summary, salt distribution using OIDA as described above would be insecure against adversaries who are able to observe the network traffic of the client for a longer time and can correlate this traffic to him, but may help in case of casual eavesdroppers or node-level adversaries who only once capture a query for a salt-protected hash.

To conclude this discussion on confidentiality of the query content, without any additional shared value between provider and client the privacy protection offered by the hash function and encrypted documents would only help against casual attackers. However, if it could be managed to share a random salt s between provider and client, dictionary attacks on the hash function would become much harder.

QC, Confidentiality of Identity, Anonymity. In Section 2.3.3, we defined a special *query confidentiality* requirement (QC), presented in a weak and strong form. QC can considered as a most highly regarded security requirement of – not necessarily privacy-fundamentalist, but cautious – individuals like *Bob Concerned*,[118] or of companies with strict information security policies against information leakage.

In the *strong* and *medium confidentiality scenario* defined above, weak QC can achieved by keying the CHF and by encryption of the retrieved document. However, for situations where the assumptions of those scenarios on key distribution do not hold, further measures could be implemented in OIDA to protect anonymity, and therefore still achieve weak QC, but this time by protecting the identity of the client.

In comparison to ONS, OIDA offers by construction better anonymity, if recursive routing is used in the underlying DHT. In theory, only the first node contacted by the client and a local ISP of the OIDA gateway may be able to see the source IP of the original query, which can betray the clients identity – if it is not obfuscated by query concentrating strategies or anonymized by onion routing discussed in Section 4.4. All other intermediate nodes as well as the final node, which answers the query, only know the address of the previous hop.

Concerning external adversaries in general, only entities able to perform global traffic analysis on large parts of the Internet – especially covering many DHT gate-

[118] Introduced in Section 2.3.3.

ways – would be capable of covering large subsets of all IOTNS users and link them to the queried hashes, as their queries leave the DHT gateway.[119]

However, as with all real-world systems, adversaries could form *beliefs* on the identities of the message sources, which can be modeled by probability distributions over the set of all possible senders, the sender anonymity set. The adversary's *uncertainty* can then be expressed by the (Shannon) *entropy* of the adversary's probability distribution over the sender anonymity set.[120]

Recently emerging research on anonymity in structured P2P systems has applied this or similar metrics to launch analyses of sender and recipient anonymity in some DHT geometries, mainly the Chord ring.[121]

Anonymity metrics based on Shannon entropy are by definition averages, or global metrics. A system can have a high anonymity value, but still single individuals may barely be protected.[122] Correspondingly, further metrics have been proposed to give a more sophisticated perspective on anonymity in a system than averaging metrics alone could provide.[123] In addition, even if only the DHT geometry without other route optimizing or proactive caching layers is analyzed, the IOTNS query distribution is still unknown, and many simulation runs would be needed to ascertain the value of an anonymity metric in several specific scenarios.[124]

Therefore, from the perspective of security engineering, this emerging field of research still lacks a set of standard measures for achieving higher anonymity in arbitrary P2P-ONS systems, which would be able to satisfy all, even individual, anonymity requirements, though ideas like limiting an adversary's node coverage, dummy traffic, the randomizing of routing, but also a fixed node in-degrees seem to play an important role. Further research toward this goal would be highly important to further analyze and strengthen anonymity in systems like OIDA.

Confidentiality of Client Location. In ONS and OIDA, the location of a client – respectively, of its resolving name server or OIDA proxy – is identifiable with a small amount of uncertainty by the query source IP address.[125] The difference is again that recursive name resolution keeps the set of potential adversaries small in

[119] Inbound query traffic from clients to the OIDA gateway would be protected by TLS, outbound traffic mixed with routed queries, but probably linkable via timing analysis.

[120] This measure for anonymity in communication systems has been proposed concurrently by Díaz et al., 2002 [41], and Serjantov and Danezis, 2003 [181].

[121] Borisov, 2005 [19]; Borisov and Waddle, 2005 [20]; Ciaccio, 2006 [32]; Ray and Zhang, 2007 [165]; O'Donnell and Vaikuntanathan, 2004, [143]. The first paper on anonymizing Chord, yet without using anonymity metrics, is Hazel and Wiley, 2002, [87].

[122] Tóth et al., 2004 [201]; Pfitzmann and Hansen, 2008 [156].

[123] Tóth et al., 2004 [201]; Clauss and Schiffner, 2006, [34].

[124] Cf. for Chord: Borisov, 2005 [19].

[125] There are public databases to query for the geographical location of clients, for example `http://www.geoiptool.com` (05.2008).

OIDA.[126]

Changing the perspective, also the EPC, especially if serial lookup is used, can function as a tracking identifier, if the corresponding object is carried by an end user and queries for it are issued from different sources over time. Similar to the EPC in ONS, the CHF value $h(e, s, r)$ could also be used as a trackable identifier in OIDA. This risk is limited by recursive routing and the option of changing salts, but offers another argument for increasing anonymity in the underlying DHT, especially in UC environments with high RFID reader density and correspondingly frequent IOTNS lookups.

Further Caveats in Client Confidentiality. Reflecting on a previously discussed indirect problem of ONS (see Section 3.3.4) – namely that even if the actual ONS query would stay confidential against eavesdroppers, a potential subsequent DNS request might not – it is important to either store only IP addresses in OIDA documents, or to include additional name resolution features in the DHT (see Section 5.7). In addition, if the two resolution phases – IOTNS and EPCIS phase – are also kept separate with OIDA, the EPCIS access, though encrypted, could give hints to external adversaries about the nature of the queries issued (see Section 2.3.3, Table 2.2).

A final problem could be the discovery of further weaknesses in established cryptographic hash functions. Research for a new standard in hash functions would help to increase client privacy, too.[127] In practical deployment, the underlying DHTs and its clients should be able to use different hash functions, for example if a new standard emerges. The currently stored data would in the case of a sudden change of the CHF have to be redistributed, which seems impractical. This could be mitigated by a time frame of using both functions in parallel until the TTL of the old data expires, during which clients issue queries using both CHFs.

5.7 OIDA Beyond ONS

OIDA can be used for more purposes than the basic EPC or OID resolution necessary for P2P-ONS. First of all, it could be used to replace DNS functionality, for example for all domain names used in OIDA documents. Regular EPCIS address documents could then contain an URL or another EPCIS service identifier, for example one usable with recent approaches for separating locators (IP addresses for routing) from identifiers in the Internet, like HIP.[128] A second document would contain the

[126] Resolving name servers have to query iteratively in DNS because of the tremendous load on the root and TLD servers that recursive resolution would generate.

[127] Cf. Burr, 2006, [26]; see also Section 5.3.1.

[128] The Host Identity Protocol is still considered experimental at the time of this writing. Main references are RFC 4423 and RFC 5205.

address data corresponding to the identifier or domain name, similar to a DNS Address RR. The first document would provide the lookup key for the second.

OIDA could also function as a meta DS, a registry for other DS. This would be implemented by storing DS addresses and service descriptions as OIDA documents. Discovery Service addresses can also be included in regular documents for retrieval of additional information on specific objects, or of objects of the same manufacturer and object class. These addresses can provide the "glue" between OIDA and future heterogeneous DS.

Depending on the requirements for EPCIS Discovery Services, OIDA itself can be used for the actual task of a DS as well, for example to let arbitrary but authorized publishers store information for particular EPCs or object classes, not only of the manufacturer as with ONS. Serial-level lookups are easily possible, with the discussed caveat on scalability problems for large IOT adoption scenarios.

Using OIDA for the actual EPCIS data is also possible, if publishers can be convinced to overcome psychological barriers, and key management for access control or a PKI covering the clients is provided.

5.8 Architecture Comparison

In this section, we summarize the preceding discussions of the main IOTNS architectures in this thesis: ONS, MONS, and OIDA. Table 5.4 gives a high-level overview if the functional, scalability, and performance requirements identified in Chapter 2 are fulfilled by each particular architecture.

Table 5.5 summarizes robustness, availability, and integrity. Finally, Table 5.6 shows instances of the confidentiality requirements, and states which architecture is able to satisfy them. In the case of client confidentiality requirements, important counter-stakeholders[129] are presented: the queried IOTNS node, i.e., a leaf server in ONS and MONS, or a DHT node in OIDA; the single or multiple roots of ONS and MONS, not applicable to OIDA; and the Internet Service Provider (ISP) or an Internet backbone operator.

5.9 Summary

OIDA decouples IOT name service tasks from the classical DNS infrastructure, which prevents an overburdening of the DNS with new applications depending on the IOT. Using a DHT for ONS will fulfill many of the requirements stated in Chapter 2. OIDA inherits the advantages of the underlying DHT architecture, which includes scalability, load distribution, redundancy, self-organization, and automatic repair

[129] Which are functional system roles in these examples, but not in general, cf. Ch. 2.

Requirement	**ONS**	**MONS**	**OIDA**
Functional Requirements			
Membership & Authorization	by environment	by environment	by environment
Flexible OID Support	no	no	✓
Single Publisher for specific OID	✓	✓	✓
Multiple Publishers (independent)	no	no	✓
Querying	✓	✓	✓
Updating	✓	✓	✓
Deleting	✓	✓	✓
Class-level Addresses	✓	✓	✓
Serial-level Addresses	✓	✓	✓
Object Information	possible via RR?	possible via RR?	✓
Scalability Requirements			
High Node Count	✓	✓	✓
High Client Count	✓	✓	✓
Medium IOT Adoption	✓	✓	✓
Large IOT Adoption – Class-level	✓	✓	✓
Large IOT Adoption – Serial-level	no	no	no
Performance Requirements			
Fast Update Propagation	no (caching)	no (caching)	✓
Low Latency	✓	✓	✓
Ultra-Low Latency	✓(caching)	✓(caching)	✓(proact. caching)
Load (Leaf Server / DHT Node)	moderate	moderate	moderate
Load (Root, TLD)	massive	massive	n/a

Table 5.4: Architecture Summary – Function, Scalability, Performance

Requirement	**ONS**	**MONS**	**OIDA**
Availability Requirements			
Robustness (Random Error)	✓	✓	✓
Robustness (Attack)	no	no	✓
Multipolarity	no	✓	✓
Integrity Requirements			
Authenticated Node Membership	ONSSEC	MONSSEC	Node PKI
Authenticated Non-Existence	ONSSEC	MONSSEC	Node PKI
Authenticated IOTNS Routing	ONSSEC	MONSSEC	Node PKI
Malicious Internal Node	possible	possible	possible
Data Integrity (End-to-End)	ONSSEC	MONSSEC	✓

Table 5.5: Architecture Summary – Availability, Integrity

Confidentiality Requirements	**ONS**	**MONS**	**OIDA**
Shared Requirements: Provider and Client			
Address-Data Confidential?	no	no	✓
Object-Data Confidential?	n/a / no	n/a / no	✓
Provider Confidentiality Requirement			
Provider Identity Confidential?	no	no	no
Client Confidentiality Requirements			
Counter-stakeholder: Queried IOTNS Node			
Query Time Confidential?	no	no	no
EPC Manager Confidential?	no	no	✓
EPC Object Class Confidential?	no	no	✓
EPC Serial (if used) Confidential?	no	no	✓
EPCIS Address Document Confidential?	no	no	✓
Source IP Confidential?	no	no	✓(not w.r.t. local gateway)
Client Anonymity?	no	no	✓(not w.r.t. local gateway)
Query Confidentiality?	no	no	✓
All Trackable Identifiers Confidential?	no	no	no: $h(e, s, r)$
Client Location Confidential?	no	no	✓(not w.r.t. local gateway)
Client Unobservability?	no	no	no
Counter-stakeholder: IOTNS Root (if not cached)			
Query Time Confidential?	no	no	n/a
EPC Manager Confidential?	no	no	n/a
EPC Object Class Confidential?	no	no	n/a
EPC Serial (if used) Confidential?	no	no	n/a
EPCIS Address Document Confidential?	✓	✓	n/a
Source IP Confidential?	no	no	n/a
Client Anonymity?	no	no	n/a
Query Confidentiality?	no	no	n/a
All Trackable Identifiers Confidential?	no	no	n/a
Client Location Confidential?	no	no	n/a
Client Unobservability?	no	no	n/a
Counter-stakeholder: Internet Backbone or ISP			
Query Time Confidential?	no	no	no
EPC Manager Confidential?	no	no	✓
EPC Object Class Confidential?	no	no	✓
EPC Serial (if used) Confidential?	no	no	✓
EPCIS Document Confidential?	no	no	✓
Source IP Confidential?	no	no	✓
Client Anonymity?	no	no	✓(not w.r.t. gateway ISP)
Query Confidentiality?	no	no	✓
All Trackable Identifiers Confidential?	no	no	no: $h(e, s, r)$
Client Location Confidential?	no	no	✓(not w.r.t. gateway ISP)
Client Unobservability?	no	no	no

Table 5.6: Architecture Summary – Confidentiality

mechanisms if nodes fail.

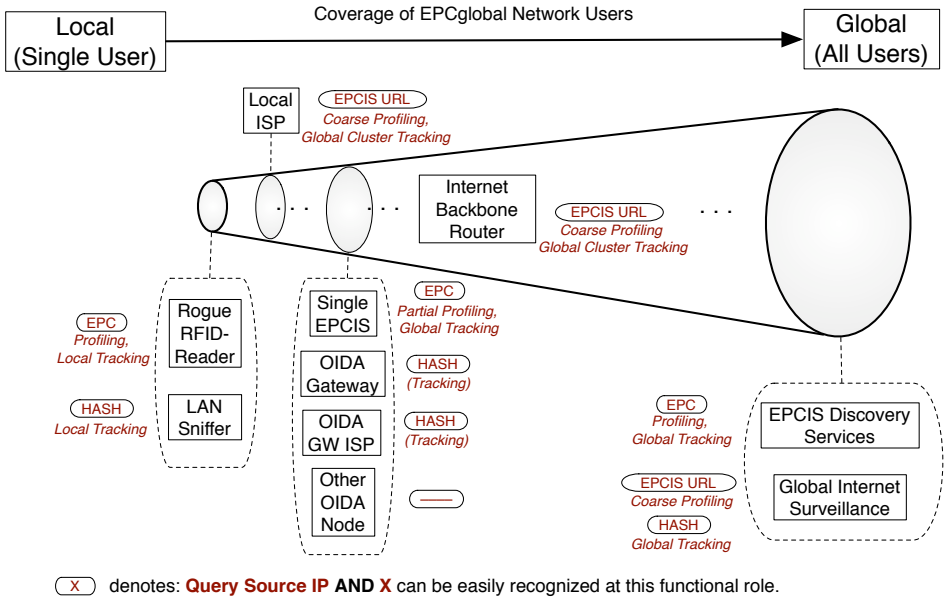


Figure 5.11: OIDA Adversary Coverage

Low service latency is possible with OIDA, and can be improved by proactive caching layers. Furthermore, robustness against random errors and denial-of-service attacks is achievable by DHT-internal or provider-controllable, easy replication mechanisms. Regarding other security requirements, access control could be implemented on the documents to satisfy a provider's confidentiality requirements.

OIDA offers enhanced confidentiality and anonymity compared to ONS, under the assumption that recursive routing is used, the gateway connections are TLS-secured, and salts are available to protect the pre-image of the CHF, as well as keys to encrypt the documents – a necessary condition for access control as well.

OIDA makes it significantly more difficult for adversaries and all functional system roles in the IOT – beyond local network boundaries – to collect (IP, EPC) pairs, or to track clients. If EPCIS Discovery Services or even EPCIS data would be integrated into OIDA, which would be possible due to the underlying document-agnostic DHT, the confidentiality situation for the whole IOT could improve even more, a trade-off with an increase of system load. See Fig. 5.11 for a change in adversary coverage if OIDA is used instead of ONS in the EPCglobal Network, compare Fig. 3.4 in Section 3.4.4.

Without appropriate key or salt distribution methods, however, client privacy can only be gradually enhanced by protecting queries and responses from casual eavesdroppers using weak and rather improvised keys. In addition, the CHF values itself may constitute a trackable identifier for those adversaries who are able to observe

the query source address. In addition, unobservability of the query is not satisfied in OIDA. To cover those cases, stronger anonymity systems or corresponding DHT anonymity enhancements may become necessary.

Chapter 6

Conclusion

This chapter summarizes the contributions of this thesis, and presents open research questions for future work in the area of IOT name services and their security.

6.1 Thesis Summary

First, this thesis presented a systematic, in-depth discussion of the functional and non-functional requirements for IOT name services, including client aspects of multilateral security, which have been neglected in the IOT standards and most literature so far.

Second, the most influential IOTNS standard *Object Naming Service* (ONS) was analyzed with respect to its security properties, discovering major shortcomings in its design. This analysis was based on the first publication in the research field of IOTNS security. Third, security enhancements to ONS were presented and discussed in depth, which could mitigate some of the identified security problems in an evolutionary way, without completely modifying the established standard. Special attention was given to MONS, a redesign of ONS to achieve multipolarity.

Finally, a new IOTNS architecture based on Distributed Hash Tables (DHT) was presented, and shown to offer better overall security than ONS while offering equivalent scalability in several possible scenarios of RFID and IOT adoption.

The implementation and testing of OIDA on the international research network PlanetLab was described, giving empirical evidence for the feasibility of P2P-ONS. A security analysis of OIDA in different scenarios for key distribution was conducted, and additional security measures and their adaptation to OIDA were discussed, including trade-offs with flexibility and performance. A comparison of ONS, MONS, and OIDA with respect to the initially identified requirements concluded the main part of the thesis.

To put the new architectures presented in this thesis in perspective, MONS and

OIDA could both enhance reliability and avoid unilateral control of the IOTNS. They can offer authenticity by – possibly distributed – authentication infrastructures like ONSSEC, a web of trust, or a distributed CA. MONS avoids single point of failure in the ONS Root; OIDA offers even more reliability by avoiding any special nodes at all and by providing flexible replication mechanisms to each publisher. OIDA enhances query confidentiality and anonymity. OIDA could also be used for some instances of EPCIS Discovery Services, which follow the name service paradigm, and could also work for actual object data (i.e., EPCIS) if encryption and access control is feasible due to an existing key distribution infrastructure.

Finally, a great potential to be investigated in future research could lie in the adoption of hybrid infrastructures – e.g., consisting of MONS Roots and regional DHTs – for guaranteed multipolarity, and improved scalability in large IOT-adoption scenarios.

6.2 Open Questions

The future will show how far and fast the process of adopting RFID and the emerging Internet of Things in supply chains will continue in future, and if a diffusion of the IOT to Ubiquitous Computing will become reality. Both developments should be continuously investigated in future research. Depending on the adoption and future application areas of the IOT, as well as the amount of RFID tags that are used in the field, the huge problem of service scalability for serial-level identifier lookup – and in general also EPCIS data management – must be tackled, which affects all architectures and system components of the IOT.

A related topic concerns IOTNS scalability and proactive caching: to conduct research on the properties of IOTNS query distributions for single EPCs, as well as for whole classes of EPCs, and what confidentiality and anonymity implications would follow from proactive caching methods if they are applied to P2P-ONS systems like OIDA. In the area of structured P2P systems, anonymity is still an emerging research area, without a final set of compelling or easy to calculate metrics, in lack also of software that might be used productively or even prototypes. Furthermore, security implications of hybrid architectures and of hierarchical or location-aware DHTs need to be studied in-depth.

A fundamental research issue for IOT security is the topic of public-key infrastructures, key distribution and revocation procedures, and their scalability for global open business and UC environments. For supply chain use of the IOT, a global PKI could be established, for example as part of the security services announced by EPCglobal. Whether such a PKI could scale to arbitrary information publishers in an extended IOT is an open problem. Related is a similar issue – the distribution of shared secrets for RFID tag access control or deactivation procedures. If the latter could be solved, perhaps even for open UC environments, the distribution of keys

to IOT clients could be established as well.

In the area of multipolarity research for the IOT and IOTNS in particular, the sharing of the Root CA functionality, e.g., by threshold cryptography, as well as its performance properties need to be investigated further. On the policy side, analysis of the practical political and administrative challenges of distributing control over the ONS is an important line for future research.

Discovery Services, once specified, will probably consist of several different designs, ranging from proper name services to whole Web services landscapes, deeply interacting with EPCIS and semantic business layers. Most performance and security requirements identified in this thesis will apply to DS as well, and iterations including additional requirements elicitation, design, implementation, and security analysis will have to be conducted in future.

Multilateral security requirements elicitation for the IOT in general will present ongoing challenges due to new and changing application areas, even more so the design of scalable and flexible architectures to satisfy them. This especially holds for the EPCIS access. If IOTNS and EPCIS phases are kept separate in future implementations of the IOT, even if OIDA is used instead of ONS, detailed information on the client and its interests can be gathered by the EPCIS provider. This could, from today's perspective, only be mitigated by using stronger measures to enforce client confidentiality like onion routing or PIR, which would result in a conflict with access control demands to enforce a publisher's confidentiality requirements.

Finally, emerging trends in Internet routing research and new naming paradigms, e.g., the separation of locators from identifiers and the Host Identification protocol (HIP), could bring about new requirements and challenges for the IOT, and its name services in particular.

Bibliography

[1] K. Albrecht and L. McIntyre. *Spychips*. Nelson Current, 2005.

[2] R. Anderson. The Eternity Service. In *Proc. Pragocrypt '96*, 1996.

[3] R. Anderson. *Security Engineering – A Guide to Building Dependable Distributed Systems*. Wiley, New York, 2001.

[4] S. Androutsellis-Theotokis and D. Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. *RFC 4033*, 2005. URL `http://www.ietf.org/rfc/rfc4033.txt`.

[6] D. Asonov and J.-C. Freytag. Almost Optimal Private Information Retrieval. In R. Dingledine and P. Syverson, editors, *Privacy Enhancing Technologies*, LNCS 2482, pages 209–223. Springer-Verlag, Berlin-Heidelberg, 2002.

[7] D. Atkins and R. Austein. Threat Analysis of the Domain Name System (DNS). *RFC 3833*, 2004. URL `http://www.ietf.org/rfc/rfc3833.txt`.

[8] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, Jan-Mar 2004.

[9] G. Avoine. Online Bibliography on Security and Privacy in RFID Systems, 2008. URL `http://www.avoine.net/rfid/`.

[10] H. Balakrishnan, M. F. Kaashoek, D. R. Karger, R. Morris, and I. Stoica. Looking up Data in P2P Systems. *Communications of the ACM*, 46(2):43–48, 2003.

[11] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish. A Layered Naming Architecture for the Internet. In *Proc. SIGCOMM '04*, pages 343–352. ACM Press, New York, 2004.

[12] M. Bauer, B. Fabian, M. Fischmann, and S. Gürses. Emerging Markets for RFID Traces. Peer-reviewed White Paper, http://arxiv.org/abs/cs.CY/0606018, 2006.

[13] I. Baumgart. P2PNS: A Secure Distributed Name Service for P2PSIP. In *Proc. 5th IEEE International Workshop on Mobile Peer-to-Peer Computing (MP2P'08) in conj. with IEEE PerCom'08, Hong Kong, China*, pages 480–485, 2008.

[14] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *Computer Communication Review*, 19(2):32–48, April 1989.

[15] O. Berthold. Effizienter Unbeobachtbarer Datenbankzugriff. In O. K. Ferstl, E. J. Sinz, S. Eckert, and T. Isselhorst, editors, *Proc. Wirtschaftsinformatik 2005*, pages 1267–1286. Physica-Verlag, Heidelberg, 2005.

[16] G. R. Blakley. Safeguarding Cryptographic Keys. In *Proc. National Computer Conference*, volume 48, pages 313–317, 1979.

[17] E.-O. Blaß, B. Fabian, M. Fischmann, and S. F. Gürses. Security in Ad-hoc and Sensor Networks. In D. Wagner and R. Wattenhofer, editors, *Algorithms for Ad-hoc and Sensor Networks*, LNCS 4621, pages 305 – 323. GI, Springer-Verlag, Berlin-Heidelberg, 2007.

[18] J. Bohn, V. Coroama, M. Langheinrich, F. Mattern, and M. Rohs. Living in a World of Smart Everyday Objects. *Journal of Human and Ecological Risk Assessment*, 10 (5):763–786, Oktober 2004.

[19] N. Borisov. *Anonymity in Structured Peer-to-Peer Networks*. PhD thesis, UC Berkeley, 2005.

[20] N. Borisov and J. Waddle. Anonymity in Structured Peer-to-Peer Networks. Technical Report UCB/CSD-05-1390, EECS Department, UC Berkeley, 2005.

[21] E. Brickell, J. Camenisch, and L. Chen. Direct Anonymous Attestation. In *Proc. 11th ACM Conference on Computer and Communications Security (CCS '04)*, pages 132–145. ACM Press, 2004.

[22] BRIDGE. BRIDGE WP02 – Requirements Document of Serial Level Lookup Service for Various Industries, August 2007. URL `http://www.bridge-project.eu/`.

[23] BRIDGE. BRIDGE WP04 – Security Analysis Report, July 2007. URL `http://www.bridge-project.eu/`.

[24] H.-J. Bullinger and M. ten Hompel, editors. *Internet der Dinge*. Springer-Verlag, Berlin-Heidelberg, 2007.

[25] M. Burmester and Y. G. Desmedt. Is Hierarchical Public-Key Certification the Next Target for Hackers? *Communications of the ACM*, 47(8):68–74, 2004.

[26] W. E. Burr. Cryptographic Hash Standards: Where Do We Go from Here? *IEEE Security and Privacy*, 4(2):88–91, March-April 2006.

[27] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Secure Routing for Structured Peer-to-peer Overlay Networks. *SIGOPS Oper. Syst. Rev.*, 36(SI): 299–314, 2002.

[28] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[29] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin. *Firewalls and Internet Security.* Addison-Wesley, 2nd edition, 2003.

[30] S. Cheung. Denial of Service against the Domain Name System. *IEEE Security and Privacy*, 4(01):40–45, 2006.

[31] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private Information Retrieval. *Journal of the ACM*, 45(6):965–981, 1998.

[32] G. Ciaccio. Improving Sender Anonymity in a Structured Overlay with Imprecise Routing. In *Proc. 6th Workshop on Privacy Enhancing Technologies (PET 2006), Cambridge, UK, 2006*, 2006.

[33] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley. Protecting Free Expression Online with Freenet. *IEEE Internet Computing*, 6(1):40–49, Jan.-Feb. 2002.

[34] S. Clauss and S. Schiffner. Structuring Anonymity Metrics. In *Proc. 2nd ACM Workshop on Digital Identity Management (DIM '06)*, pages 55–62. ACM Press, New York, 2006.

[35] Common Criteria. Common Criteria for Information Technology Security Evaluation, Version 3.1, September 2006. URL `http://www.commoncriteriaportal.org/public/expert/`.

[36] C. Cortes, K. Fisher, D. Pregibon, A. Rogers, and F. Smith. Hancock: A Language for Analyzing Transactional Data Streams. *ACM Transactions on Programming Languages and Systems*, 26(2):301–338, 2004.

[37] R. Cox, A. Muthitacharoen, and R. Morris. Serving DNS Using a Peer-to-Peer Lookup Service. In *Rev. Papers 1st International Workshop on Peer-to-Peer Systems (IPTPS '01)*, LNCS 2429, pages 155–165. Springer-Verlag, Berlin-Heidelberg, 2002.

[38] L. F. Cranor. 'I didn't buy it for myself' – Privacy and E-Commerce Personalization. In *Proc. 2003 ACM Workshop on Privacy in the Electronic Society (WPES '03)*, pages 111–117. ACM Press, New York, 2003.

[39] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-Area Cooperative Storage with CFS. In *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP '01), Chateau Lake Louise, Banff, Canada*, October 2001.

[40] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. Anderson. Sybil-Resistant DHT Routing. In *Proc. 10th European Symposium on Research in Computer Security (ESORICS 2005)*, LNCS 3679, pages 305–318. Springer-Verlag, Berlin-Heidelberg, 2005.

[41] C. Díaz, J. Claessens, S. Seys, and B. Preneel. Information Theory and Anonymity. In *Proc. 23rd Symposium on Information Theory in the Benelux, Louvain la Neuve, Belgium*, May 2002.

[42] T. Diekmann, A. Melski, and M. Schumann. Data-on-Network vs. Data-on-Tag: Managing Data in Complex RFID Environments. *Proc. 40th Hawaii International Conference on System Sciences (HICSS '07)*, 2007.

[43] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol. *RFC 4346*, 2006. URL `http://www.ietf.org/rfc/rfc4346.txt`.

[44] R. Dingledine, M. J. Freedman, and D. Molnar. The Free Haven Project: Distributed Anonymous Storage Service. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Proc. Workshop on Design Issues in Anonymity and Unobservability, Berkeley, USA, 2000*, LNCS 2009, pages 67 – 95. Springer-Verlag, Berlin-Heidelberg, 2001.

[45] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proc. 13th USENIX Security Symposium*, 2004.

[46] Y. Doi. DNS Meets DHT: Treating Massive ID Resolution Using DNS over DHT. In *Proc. 2005 Symposium on Applications and the Internet (SAINT '05)*, pages 9–15, 2005.

[47] J. R. Douceur. The Sybil Attack. In P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, editors, *Rev. Papers 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, LNCS 2429, pages 251 – 260. Springer-Verlag, Berlin-Heidelberg, 2002.

[48] D. Eastlake and P. Jones. US Secure Hash Algorithm 1 (SHA1). *RFC 3174*, 2001. URL `http://www.ietf.org/rfc/rfc3174.txt`.

[49] EPCglobal. The EPCglobal Network: Overview of Design, Benefits, Security, 2004. URL `http://www.epcglobalinc.org`.

[50] EPCglobal. Implementation of the EPCglobal Network Root ONS. EPCglobal Position Paper, November 2005. URL `http://www.epcglobalinc.org/`.

[51] EPCglobal. EPCglobal Tag Data Standards – Version 1.3.1, September 2007. URL `http://www.epcglobalinc.org/standards/`.

[52] EPCglobal. EPC Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID – Version 1.1.0, October 2007. URL `http://www.epcglobalinc.org/standards/`.

[53] EPCglobal. The EPCglobal Architecture Framework – Version 1.2, September 2007. URL `http://www.epcglobalinc.org/standards/`.

[54] EPCglobal. EPCglobal Certificate Profile – Version 1.0.1, May 2008. URL `http://www.epcglobalinc.org/standards/`.

[55] S. Evdokimov, B. Fabian, and O. Günther. Multipolarity for the Object Naming Service. In *Proc. Internet of Things (IOT 2008), Zurich, Switzerland, 2008*, LNCS 4952, pages 1–18. Springer-Verlag, Berlin-Heidelberg, 2008.

[56] B. Fabian. Physical Intrusion Detection Using RFID. In U. Flegel, editor, *Proc. First GI SIG SIDAR Graduate Workshop on Reactive Security (SPRING), Technical Report SR-2006-01*. GI SIG SIDAR, 2006.

[57] B. Fabian and O. Günther. Distributed ONS and Its Impact on Privacy. In *Proc. IEEE International Conference on Communications (IEEE ICC 2007), Glasgow*, 2007.

[58] B. Fabian and O. Günther. Security Challenges of the EPCglobal Network. *Communications of the ACM*, 2009.

[59] B. Fabian and M. Hansen. Technische und Organisatorische Lösungen für Informationelle Sicherheit und Selbstbestimmung im Ubiquitous Computing. In *Technikfolgen-Abschätzung Ubiquitäres Computing und Informationelle Selbstbestimmung (TAUCIS)*, chapter 7, pages 242–320. BMBF, 2006. URL `http://www.taucis.de`.

[60] B. Fabian and M. Hansen. Technische Grundlagen des Ubiquitous Computing. In *Technikfolgen-Abschätzung Ubiquitäres Computing und Informationelle Selbstbestimmung (TAUCIS)*, chapter 1, pages 11–44. BMBF, 2006. URL `http://www.taucis.de`.

[61] B. Fabian and M. Hansen. Anwendungen des Ubiquitous Computing. In *Technikfolgen-Abschätzung Ubiquitäres Computing und Informationelle Selbstbestimmung (TAUCIS)*, chapter 2, pages 45–62. BMBF, 2006. URL `http://www.taucis.de`.

[62] B. Fabian, O. Günther, and S. Spiekermann. Security Analysis of the Object Name Service. In *Proc. 1st IEEE Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2005), in conj. with IEEE ICPS 2005, Santorini, Greece*, pages 71–76, 2005.

[63] B. Fabian, S. Gürses, A. Kuzmanovski, and T. Santen. Confidentiality in Pervasive Systems – Relating Requirements to Information Inference. White Paper, 2006.

[64] B. Fabian, S. Gürses, M. Heisel, A. Kuzmanovski, T. Santen, and H. Schmidt. A Comparison of Security Requirements Engineering Methods. In Submission, 2009.

[65] K. Finkenzeller. *RFID-Handbuch - Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten.* Carl Hanser Verlag, München, 4th edition, August 2006.

[66] M. Fischmann. Modelling Reputation-Based Resource Pooling in P2P Systems. In *Proc. 1st International Conference on Scalable Information Systems (INFOSCALE '06)*. IEEE Press, 2006.

[67] M. E. Fiuczynski. PlanetLab: Overview, History, and Future Directions. *ACM SIGOPS Operating Systems Review*, 40(1):6–10, 2006.

[68] E. Fleisch and F. Mattern, editors. *Das Internet der Dinge – Ubiquitous Computing und RFID in der Praxis.* Springer-Verlag, Berlin-Heidelberg, 2005.

[69] C. Floerkemeier, R. Schneider, and M. Langheinrich. Scanning with a Purpose – Supporting the Fair Information Principles in RFID Protocols. In *Proc. 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, Tokyo, Japan, 2004.

[70] A. Friedlander, A. Mankin, W. D. Maughan, and S. D. Crocker. DNSSEC: A Protocol toward Securing the Internet Infrastructure. *Communications of the ACM*, 50(6):44–50, 2007.

[71] S. Garfinkel and B. Rosenberg, editors. *RFID Applications, Security, and Privacy.* Addison-Wesley, 2005.

[72] S. Garfinkel, A. Juels, and R. Pappu. RFID Privacy: An Overview of Problems and Proposed Solutions. *IEEE Security and Privacy*, 3(3):34–43, May-June 2005.

[73] S. L. Garfinkel. An RFID Bill of Rights. *Technology Review*, October 2002. URL `http://www.technologyreview.com/articles/02/10/garfinkel1002.asp`.

[74] D. Geer. Malicious Bots Threaten Network Security. *IEEE Computer*, 38(1):18–20, January 2005.

[75] A. Ghodsi. *Distributed k-ary System: Algorithms for Distributed Hash Tables.* PhD thesis, KTH Stockholm, Sweden, 2006. URL `http://eprints.sics.se/516/01/dks.pdf`.

[76] S. Gibbard. Geographic Implications of DNS Infrastructure Distribution. *The Internet Protocol Journal*, 10(1):12–24, 2007.

[77] H. Gilbert and H. Handschuh. Security Analysis of SHA-256 and Sisters. In *Rev. Papers from 10th Annual International Workshop on Selected Areas in Cryptography (SAC 2003), Ottawa, Canada*, LNCS 3006, pages 175–193. Springer-Verlag, Berlin-Heidelberg, 2004.

[78] D. Gollmann. *Computer Security.* Wiley, Chichester, 2nd edition, 2006.

[79] S. Guha and P. Francis. Identity Trail: Covert Surveillance Using DNS. In *Proc. PET 2007*, LNCS 4776, pages 153–166. Springer-Verlag, Berlin-Heidelberg, 2007.

[80] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The Impact of DHT Routing Geometry on Resilience and Proximity. In *Proc. ACM SIGCOMM 2003, Karsruhe, Germany*, pages 381–394. ACM Press, New York, 2003.

[81] O. Günther and S. Spiekermann. RFID and the Perception of Control: The Consumer's View. *Communications of the ACM*, 48(9):73–76, September 2005.

[82] S. Gürses and T. Santen. Contextualizing Security Goals – A Method for Multilateral Security Requirements Elicitation. In J. Dittmann, editor, *Proc. Sicherheit 2006 – Schutz und Zuverlässigkeit*, LNI, pages 42–53. Gesellschaft für Informatik, 2006.

[83] S. Gürses, B. Berendt, and T. Santen. Multilateral Security Requirements Analysis for Preserving Privacy in Ubiquitous Environments. In B. Berendt and E. Menasalvas, editors, *Proc. Workshop on Ubiquitous Knowledge Discovery for Users (UKDU '06)*, pages 51–64, 2006.

[84] J. Han, A. Jain, M. Luk, and A. Perrig. Don't Sweat Your Privacy: Using Humidity to Detect Human Presence. In *Proc. 5th International Workshop on Privacy in UbiComp, September 2007, Innsbruck, Austria*, 2007.

[85] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer, New York, 2004.

[86] M. Harrison. EPC Information Service (EPCIS). In *Proc. Auto-ID Labs Research Workshop*. Auto-ID Labs Zürich, 2004. URL `http://www.m-lab.ch/auto-id/SwissReWorkshop/papers/EPCinformationService.pdf`.

[87] S. Hazel and B. Wiley. Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems. In *Proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, 2002.

[88] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The Gator Tech Smart House: A Programmable Pervasive Space. *IEEE Computer*, pages 50–60, March 2005.

[89] J. R. Hind, J. M. Mathewson, and M. L. Peters. Identification and Tracking of Persons Using RFID-Tagged Items. US Patent Application 20020165758, 2001.

[90] Honeynet Project. Know Your Enemy: Fast-Flux Service Networks, July 2007. URL `http://www.honeynet.org/papers/ff/`.

[91] L. Huang. Distributed DNS Implementation in IPv6. Internet Draft, April 2007. URL `http://tools.ietf.org/html/draft-licanhuang-dnsop-distributeddns-03.txt`.

[92] ICANN. Root Server Attack on 6 February 2007. ICANN Factsheet, 2007. URL `http://www.icann.org/announcements/factsheet-dns-attack-08mar07.pdf`.

[93] A. Iliev and S. W. Smith. Protecting Client Privacy with Trusted Computing at the Server. *IEEE Security and Privacy*, 3(2):20 – 28, March – April 2005.

[94] ISO/IEC 13335. Information Technology – Security Techniques – Management of Information and Communications Technology Security – Part 1: Concepts and Models for Information and Communications Technology Security Management (ISO/IEC 13335-1:2004), 2004.

[95] Y. Ivanov, C. Wren, A. Sorokin, and I. Kaur. Visualizing the History of Living Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1153–1160, Nov.-Dec. 2007.

[96] M. Jackson. *Problem Frames. Analyzing and Structuring Software Development Problems*. Addison-Wesley, Boston, 2001.

[97] R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley, Chichester, 1991.

[98] A. Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, February 2006.

[99] A. Juels, R. Pappu, and B. Parno. Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In *Proc. 17th USENIX Security Symposium 2008, San Jose, USA*, 2008.

[100] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. In *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement (IMW '01), San Francisco, California, USA*, pages 153–167. ACM Press, New York, NY, USA, 2001.

[101] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. *IEEE/ACM Transactions on Networking*, 10(5):589–603, 2002.

[102] M. F. Kaashoek and D. R. Karger. Koorde: A Simple Degree-optimal Distributed Hash Table. In *Proc. 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.

[103] B. Kaliski. PKCS 5: Password-Based Cryptography Specification Version 2.0. *RFC 2898*, 2000. URL `http://www.ietf.org/rfc/rfc2898.txt`.

[104] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis. A Fair Solution to DNS Amplification Attacks. *Proc. 2nd International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007)*, pages 38–47, 2007.

[105] D. Kaminsky. Explorations in Namespace: White-hat Hacking Across the Domain Name System. *Communications of the ACM*, 49(6):62–69, 2006.

[106] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent Hashing and Random Trees. In *Proc. 29th Annual ACM Symposium on Theory of Computing (STOC '97), El Paso, Texas, USA*, pages 654–663. ACM Press, New York, NY, USA, 1997.

[107] G. Karjoth and P. Moskowitz. Disabling RFID Tags with Visible Confirmation: Clipped Tags Are Silenced. In *Proc. Workshop on Privacy in the Electronic Society (WPES 2005), Alexandria, Virginia, USA*. ACM Press, New York, 2005.

[108] K. Kaya and A. A. Selcuk. Threshold Cyptography Based on Asmuth-Bloom Secret Sharing. *Information Sciences*, 177(19):4148–4160, 2007.

[109] D. Kesdogan, M. Borning, and M. Schmeink. Unobservable Surfing on the World Wide Web: Is Private Information Retrieval an Alternative to the MIX Based Approach? In R. Dingledine and P. Syverson, editors, *Privacy Enhancing Technologies (PET 2002)*, LNCS 2482, pages 224–238. Springer-Verlag, Berlin-Heidelberg, 2003.

[110] A. Klein. BIND 8 DNS Cache Poisoning, July–August 2007. URL `http://www.trusteer.com/docs/bind8dns.html`.

[111] A. Klein. BIND 9 DNS Cache Poisoning, July 2007. URL `http://www.trusteer.com/docs/bind9dns.html`.

[112] J. M. Kleinberg. Navigation in a Small World. *Nature*, 406(6798):845–845, 2000.

[113] D. M. Konidala, W.-S. Kim, and K. Kim. Security Assessment of EPCglobal Architecture Framework, 2006. URL `http://www.autoidlabs.org/uploads/media/AUTOIDLABS-WP-SWNET-017.pdf`.

[114] B. Kuerbis and M. Mueller. Securing the Root: A Proposal for Distributing Signing Authority. Paper IGP07-002, 2007. URL `http://www.internetgovernance.org/pdf/SecuringTheRoot.pdf`.

[115] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse. Discovery Service Design in the EPCglobal Network – Towards Full Supply Chain Visibility. In *Proc. Internet of Things (IOT 2008), Zurich, Switzerland, 2008*, LNCS 4952, pages 19–34. Springer-Verlag, Berlin-Heidelberg, 2008.

[116] V. T. Lam, S. Antonatos, P. Akritidis, and K. G. Anagnostakis. Puppetnets: Misusing Web Browsers as a Distributed Attack Infrastructure. In *Proc. ACM Conference on Computer and Communications Security (CCS '06)*, pages 221–234, 2006.

[117] M. Langheinrich and R. Marti. Practical Minimalist Cryptography for RFID Privacy. *IEEE Systems Journal, Special Issue on RFID Technology*, 1(2):115–128, Dec. 2007.

[118] G. Lawton. Stronger Domain Name System Thwarts Root-Server Attacks. *IEEE Computer*, 40(5):14–17, 2007.

[119] K. S. Leong, M. L. Ng, et al. EPC Network Architecture. In *Auto-ID Labs Research Workshop*. Auto-ID Labs Zürich, 2004. URL `http://www.m-lab.ch/auto-id/SwissReWorkshop/agenda.html`.

[120] J. Leyden. Homeland Security Grabs for Net's Master Keys. The Register, 3 April 2007, 2007. URL `http://www.theregister.co.uk/2007/04/03/dns_master_key_controversy`.

[121] R. Liston, S. Srinivasan, and E. Zegura. Diversity in DNS Performance Measures. In *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurement (IMW '02), Marseille, France*, pages 19–31. ACM Press, New York, 2002.

[122] C. Liu and P. Albitz. *DNS and BIND.* O'Reilly & Associates, 5th edition, 2006.

[123] S. Liu, F. Wang, and P. Liu. Integrated RFID Data Modeling: An Approach for Querying Physical Objects in Pervasive Computing. In *Proc. 15th ACM International Conference on Information and Knowledge Management (CIKM '06)*, pages 822–823, 2006.

[124] D. Loguinov, J. Casas, and X. Wang. Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience. *IEEE/ACM Transactions on Networking*, 13(5):1107–1120, 2005.

[125] J. Lopez, R. Oppliger, and G. Pernul. Why Have Public Key Infrastructures Failed so far? *Internet Research*, 15(5), October 2005.

[126] P. Loshin. *IPv6, Theory, Protocol and Practice*. Elsevier, San Francisco, 2004.

[127] W. Mao. *Modern Cryptography – Theory & Practice*. Prentice Hall / Pearson Education, Upper Saddle River, 2004.

[128] F. Mattern. Vom Verschwinden des Computers – Die Vision des Ubiquitous Computing. In F. Mattern, editor, *Total Vernetzt*, pages 1–41. Springer-Verlag, 2003.

[129] M. Mealling. EPCglobal Object Naming Service (ONS) 1.0, 2005. URL `http://www.epcglobalinc.org/standards/`.

[130] M. Mealling and R. Daniel. The Naming Authority Pointer (NAPTR) DNS Resource Record. *RFC 2915*, September 2000. URL `http://www.ietf.org/rfc/rfc2915.txt`.

[131] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[132] P. Mockapetris. Domain Names - Concepts and Facilities. *RFC 1034*, 1987. URL `http://www.ietf.org/rfc/rfc1034.txt`.

[133] P. Mockapetris. Domain Names - Implementation and Specification. *RFC 1035*, 1987. URL `http://www.ietf.org/rfc/rfc1035.txt`.

[134] R. Morris and K. Thompson. Password Security: A Case History. *Communications of the ACM*, 22(11):594–597, 1979.

[135] S. J. Murdoch. Covert Channel Vulnerabilities in Anonymity Systems. Technical Report UCAM-CL-TR-706, University of Cambridge, Computer Laboratory, Dec. 2007.

[136] S. J. Murdoch and G. Danezis. Low-Cost Traffic Analysis of Tor. In *Proc. 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

[137] S. J. Murdoch and P. Zielinski. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *Proc. PET 2007*, 2007.

[138] R. M. Needham. Denial of Service. In *Proc. 1st ACM Conference on Computer and Communications Security (CCS '93), Fairfax, Virginia, USA*, pages 151–153. ACM Press, New York, 1993.

[139] M. Newman. Power Laws, Pareto Distributions and Zipf's Law. *Contemporary Physics*, 46:323–351, Sept. 2005.

[140] NIST. Secure Hash Standard. National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 180-1, April 1993.

[141] NIST. Advanced Encryption Standard (AES). National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 197, November 2001.

[142] A. Odlyzko. Privacy, Economics, and Price Discrimination on the Internet. In *Proc. 5th International Conference on Electronic Commerce (ICEC '03)*, pages 355–366. ACM Press, New York, 2003.

[143] C. W. O'Donnell and V. Vaikuntanathan. Information Leak in the Chord Lookup Protocol. In *Proc. 4th International Conference on Peer-to-Peer Computing (P2P'04), Zurich, Switzerland*, pages 28–35. IEEE CS, Washington, 2004.

[144] P. Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. *Proc. Advances in Cryptology (CRYPTO 2003)*, pages 617–630, 2003.

[145] R. Oppliger, R. Hauser, and D. Basin. SSL/TLS Session-Aware User Authentication. *IEEE Computer*, 41(3):59–65, 2008.

[146] A. Ozment and S. E. Schechter. Bootstrapping the Adoption of Internet Security Protocols. In *Proc. 5th Workshop on the Economics of Information Security (WEIS 2006)*, 2006.

[147] A. Ozment, S. E. Schechter, and R. Dhamija. Web Sites Should Not Need to Rely on Users to Secure Communications. In *Proc. W3C Workshop on Transparency and Usability of Web Authentication*, 2006.

[148] J. Pang, J. Hendricks, A. Akella, R. D. Prisco, B. Maggs, and S. Seshan. Availability, Usage, and Deployment Characteristics of the Domain Name System. In *Proc. 4th ACM SIGCOMM Conference on Internet Measurement (IMC '04), Taormina, Sicily, Italy*, pages 1–14. ACM Press, New York, 2004.

[149] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang. Impact of Configuration Errors on DNS Robustness. In *Proc. ACM SIGCOMM '04, Portland, Oregon, USA*, pages 319–330. ACM Press, New York, 2004.

[150] V. Pappas, D. Massey, A. Terzis, and L. Zhang. A Comparative Study of the DNS Design with DHT-Based Alternatives. *Proc. INFOCOM 2006*, pages 1–13, 2006.

[151] K. Park, V. S. Pai, L. Peterson, and Z. Wang. CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups. In *Proc. 6th Symposium on Operating Systems Design and Implementation (OSDI 2004), San Francisco, CA, USA*, 2004.

[152] T. Peng, C. Leckie, and K. Ramamohanarao. Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems. *ACM Computing Surveys*, 39(1):3, 2007.

[153] D. Peppers, M. Rogers, and B. Dorf. *The One to One Fieldbook*. Capstone Publishing Ltd, Oxford, 1999.

[154] L. Peterson and V. S. Pai. Experience-Driven Experimental Systems Research. *Communications of the ACM*, 50(11):38–44, November 2007.

[155] L. Peterson and T. Roscoe. The Design Principles of PlanetLab. *SIGOPS Oper. Syst. Rev.*, 40(1):11–16, 2006.

[156] A. Pfitzmann and M. Hansen. Anonymity, Unobservability, and Pseudonymity: A Consolidated Proposal for Terminology. Draft, 2008. URL `http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf`.

[157] B. J. Pine II, B. Victor, and A. C. Boynton. Making Mass Customization Work. *Harvard Business Review*, pages 108–119, September-October 1993.

[158] PITAC. Cyber Security – A Crisis of Prioritization, February 2005. URL `http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf`.

[159] L. Poole and V. S. Pai. ConfiDNS: Leveraging Scale and History to Improve DNS Security. In *Proc. 3rd Workshop on Real, Large Distributed Systems (WORLDS 2006), Seattle, WA, USA*, 2006.

[160] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The Ghost In The Browser: Analysis of Web-based Malware. In *Proc. 1st Workshop on Hot Topics in Understanding Botnets April 10, 2007, Cambridge, MA*, 2007.

[161] V. Ramasubramanian and E. G. Sirer. The Design and Implementation of a Next Generation Name Service for the Internet. In *Proc. ACM SIGCOMM '04, Portland, Oregon, USA*, pages 331–342. ACM Press, New York, 2004.

[162] V. Ramasubramanian and E. G. Sirer. Beehive: O(1) Lookup Performance for Power-Law Query Distributions in Peer-to-Peer Overlays. In *Proc. 1st Symposium on Networked Systems Design and Implementation (NSDI'04), San Francisco, USA*. USENIX Association, 2004.

[163] K. Rannenberg, A. Pfitzmann, and G. Müller. IT Security and Multilateral Security. In G. Müller and K. Rannenberg, editors, *Multilateral Security in Communications – Technology, Infrastructure, Economy*, pages 21–29. Addison-Wesley, 1999.

[164] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proc. ACM SIGCOMM '01*, pages 161–172. ACM Press, New York, 2001.

[165] S. Ray and Z. Zhang. An Information-Theoretic Framework for Analyzing Leak of Privacy in Distributed Hash Tables. In *Proc. 7th IEEE International Conference on Peer-to-Peer Computing (P2P 2007), Galway, Ireland*, 2007.

[166] E. Rescorla and N. Modadugu. Datagram Transport Layer Security. *RFC 4347*, 2006. URL `http://www.ietf.org/rfc/rfc4347.txt`.

[167] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz. Maintenance-Free Global Data Storage. *IEEE Internet Computing*, 5(5):40–49, 2001.

[168] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling Churn in a DHT. In *Proc. USENIX Annual Technical Conference (ATEC '04), Boston, MA, USA*, pages 127–140. USENIX Association, Berkeley, 2004.

[169] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. OpenDHT: A Public DHT Service and Its Uses. In *Proc. ACM SIGCOMM '05, Philadelphia, Pennsylvania, USA*, pages 73–84. ACM Press, New York, 2005.

[170] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. The Evolution of RFID Security. *IEEE Pervasive Computing*, 5(1):62–69, 2006.

[171] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. Is Your Cat Infected with a Computer Virus? In *Proc. 4th IEEE Intl. Conf. on Pervasive Computing and Communications (PerCom 2006), Pisa, Italy*, 2006.

[172] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[173] E. M. Rogers. *Diffusion of Innovations.* Free Press, New York, 5th edition, 2003.

[174] M. Rothensee. User Acceptance of the Intelligent Fridge: Empirical Results from a Simulation. In *Proc. Internet of Things (IOT 2008), Zurich, Switzerland, 2008*, LNCS 4952, pages 123–139. Springer-Verlag, Berlin-Heidelberg, 2008.

[175] P. Rotter. A Framework for Assessing RFID System Security and Privacy Risks. *IEEE Pervasive Computing*, 7(2):70–77, 2008.

[176] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In R. Guerraoui, editor, *Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware '01), Heidelberg, Germany*, LNCS 2218, pages 329–350. Springer-Verlag, 2001.

[177] A. Salamon. DNS-Related RFCs, 2008. URL `http://www.dns.net/dnsrd/rfc/`.

[178] T. Santen. Stepwise Development of Secure Systems. In J. Górski, editor, *Proc. 25th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2006). Gdansk, Poland*, LNCS 4166, pages 142–155. Springer, 2006.

[179] J. B. Schafer, J. A. Konstan, and J. Riedi. Recommender Systems in E-Commerce. In *Proc. ACM Conference on Electronic Commerce*, pages 158–166, 1999.

[180] B. Schneier. Attack Trees. *Dr. Dobb's Journal*, December 1999. URL `http://www.ddj.com/184411129`.

[181] A. Serjantov and G. Danezis. Towards an Information Theoretic Metric for Anonymity. In P. Syverson and R. Dingledine, editors, *Proc. Privacy Enhancing Technologies 2002*, LNCS 2482, pages 259–263, 2003.

[182] A. Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.

[183] D.-H. Shih, P.-L. Sun, and B. Lin. Securing Industry-wide EPCglobal Network with WS-Security. *Industrial Management and Data Systems*, 105(7):972–996, Sep 2005.

[184] R. Shirey. Internet Security Glossary. *RFC 2828*, May 2000. URL `http://www.ietf.org/rfc/rfc2828.txt`.

[185] V. Shoup. Practical Threshold Signatures. In *Proc. EUROCRYPT 2000, Bruges, Belgium*, volume LNCS 1807, pages 207–220. Springer-Verlag, Berlin-Heidelberg, 2000.

[186] A. Singh, M. Castro, P. Druschel, and A. Rowstron. Defending against Eclipse Attacks on Overlay Networks. In *Proc. 11th ACM SIGOPS European Workshop, Leuven, Belgium*, page 21. ACM Press, New York, 2004.

[187] E. Sit and R. Morris. Security Considerations for Peer-to-Peer Distributed Hash Tables. In P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, editors, *Rev. Papers 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, LNCS 2429, pages 261–269. Springer-Verlag, Berlin-Heidelberg, 2002.

[188] I. Sommerville. *Software Engineering*. Addison Wesley, 7th edition, 2004.

[189] J. Song and H. Kim. The RFID Middleware System Supporting Context-aware Access Control Service. In *Proc. 8th International Conference on Advanced Communication Technology (ICACT 2006)*, 2006.

[190] J. Song, T. Kim, S. Lee, and H. Kim. Security Enhanced RFID Middleware System. *Proc. World Academy of Science, Engineering and Technology (PWASET)*, December 2005.

[191] S. Spiekermann and H. Ziekow. RFID: A 7-point Plan to Ensure Privacy. In *Proc. 13th European Conference on Information Systems (ECIS)*, Regensburg, May 2005.

[192] F. Stajano. *Security for Ubiquitous Computing*. Wiley, Chichester, 2002.

[193] R. Steinmetz and K. Wehrle, editors. *Peer-to-Peer Systems and Applications*. LNCS 3485. Springer-Verlag, Berlin-Heidelberg, 2005.

[194] W. R. Stevens. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Boston, 1994.

[195] J. Stewart. DNS Cache Poisoning – The Next Generation, 2007. URL `http://www.secureworks.com/research/articles/dns-cache-poisoning/`.

[196] I. Stoica, R. Morris, D. Karger, M. Kaashock, and H. Balakrishman. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. In *Proc. ACM SIGCOMM 2001, San Diego, California*, 2001.

[197] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, 2003.

[198] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous Connections and Onion Routing. In *Proc. 1997 IEEE Symposium on Security and Privacy (SP '97)*. IEEE CS, Washington, 1997.

[199] A. S. Tanenbaum. *Computer Networks*. Prentice Hall / Pearson Education, Upper Saddle River, 4th edition, 2003.

[200] H. P. Thadakamalla, S. R. T. Kumara, and R. Albert. Complexity and Large-Scale Networks. In A. R. Ravindran, editor, *Operations Research and Management Science Handbook*, chapter 11. CRC Press, Boca Raton, 2007.

[201] G. Tóth, Z. Hornák, and F. Vajda. Measuring Anonymity Revisited. In S. Liimatainen and T. Virtanen, editors, *Proc. 9th Nordic Workshop on Secure IT Systems*, pages 85–90, Espoo, Finland, November 2004.

[202] Y. Uo, S. Suzuki, et al. Name Service on the EPC Network. *Proc. Auto-ID Labs Research Workshop*, 2004. URL `http://www.m-lab.ch/auto-id/SwissReWorkshop/papers/NameServiceOnTheEPCnetwork.pdf`.

[203] VeriSign. The EPCglobal Network: Enhancing the Supply Chain. VeriSign White Paper, 2005.

[204] P. Vixie. DNS and BIND Security Issues. In *Proc. 5th USENIX Security Symposium (SSYM'95), Salt Lake City, USA*, page 19, 1995.

[205] M. Waldman, A. Rubin, and L. Cranor. Publius: A Robust, Tamper-Evident, Censorship-Resistant and Source-Anonymous Web Publishing System. In *Proc. 9th USENIX Security Symposium, Denver, USA*, pages 59–72, 2000.

[206] S. F. Wamba and H. Boeck. Enhancing Information Flow in a Retail Supply Chain Using RFID and the EPC Network: A Proof-of-Concept Approach. *Journal of Theoretical and Applied Electronic Commerce Research*, 3(1):92–105, April 2008.

[207] S. F. Wamba, L. A. Lefebvre, and E. Lefebvre. Enabling Intelligent B-to-B eCommerce Supply Chain Management Using RFID and the EPC Network. In *Proc. 8th International Conference on Electronic Commerce (ICEC '06)*, pages 281–288, 2006.

[208] L. Wang, K. Park, R. Pang, V. S. Pai, and L. Peterson. Reliability and Security in the CoDeeN Content Distribution Network. In *Proc. USENIX 2004 Annual Technical Conference, Boston, USA*, 2004.

[209] X. Wang, Y. L. Yin, and H. Yu. Finding Collisions in the Full SHA-1. In *Advances in Cryptology – CRYPTO 2005*, LNCS 3621, pages 17–36, 2005.

[210] D. J. Watts. *Small Worlds*. Princeton University Press, Princeton, 1999.

[211] S. A. Weis. Security and Privacy in Radio-Frequency Identification Devices. Master's thesis, Massachusetts Institute of Technology, 2003.

[212] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In D. Hutter et al., editors, *Proc. Security in Pervasive Computing (SPC 2003), Boppard, Germany*, LNCS 2802, pages 201–212. Springer-Verlag, Berlin-Heidelberg, 2004.

[213] M. Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):66–75, 1991.

[214] D. Wessels. Is Your Caching Resolver Polluting the Internet? In *Proc. ACM SIGCOMM Workshop on Network Troubleshooting (NetT '04), Portland, USA*, pages 271–276. ACM Press, New York, 2004.

[215] Wikipedia. Online Encyclopedia, 2008. URL `http://www.wikipedia.org/`.

[216] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems. *ACM Transactions on Information and System Security*, 7(4):489–522, 2004.

[217] Y. Zhu and X. Yang. Implications of Neighbor Selection on DHT Overlays. *Proc. 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2006)*, pages 197–206, 2006.

[218] P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1995.

Appendix A

OIDA Bamboo Configuration

The `oida.cfg` configuration file (host `planetlab1.wiwi.hu-berlin.de`) for the OIDA prototype using the Bamboo DHT on Planetlab, cf. Section 5.4.

```
<sandstorm>
    <global>
        <initargs>
            node_id 141.20.103.210:15941
        </initargs>
    </global>

    <stages>
        <Network>
            class bamboo.network.Network
            <initargs>
#                mac_key_file /home/bamboo/mac.key
            </initargs>
        </Network>

        <Rpc>
            class bamboo.lss.Rpc
            <initargs>
            </initargs>
        </Rpc>

        <Router>
            class bamboo.router.Router
            <initargs>
                        gateway_count  8
                        gateway_0      planetlab1.wiwi.hu-berlin.de:15941
                        gateway_1      planet1.scs.cs.nyu.edu:15941
                        gateway_2      planetlab2.hiit.fi:15941
                        gateway_3      planetlab1.itwm.fhg.de:15941
                        gateway_4      mars.planetlab.haw-hamburg.de:15941
                        gateway_5      planetlab2.wiwi.hu-berlin.de:15941
                        gateway_6      planet1.zib.de:15941
                        gateway_7      planet2.zib.de:15941
                leaf_set_size  4
                digit_values   2
                immediate_join true
            </initargs>
```

```
38          </Router>
39
40          <DataManager>
41              class bamboo.dmgr.DataManager
42              <initargs>
43                  required_acks  2
44              </initargs>
45          </DataManager>
46
47          <StorageManager>
48              class bamboo.db.StorageManager
49              <initargs>
50                  homedir         /home/huberlin_oida/oida/store/store-15941
51              </initargs>
52          </StorageManager>
53
54          <Dht>
55              class bamboo.dht.Dht
56              <initargs>
57                  storage_manager_stage StorageManager
58                  min_replica_count     1
59              </initargs>
60          </Dht>
61
62          <Gateway>
63              class bamboo.dht.Gateway
64              <initargs>
65                  port            15943
66              </initargs>
67          </Gateway>
68
69          <WebInterface>
70              class bamboo.www.WebInterface
71              <initargs>
72                  storage_manager_stage StorageManager
73              </initargs>
74          </WebInterface>
75
76          <Vivaldi>
77              class bamboo.vivaldi.Vivaldi
78              <initargs>
79                vc_type           2.5d
80                generate_pings    true
81                eavesdrop_pings   false
82                use_reverse_ping  true
83                ping_period       10000
84                version           1
85              </initargs>
86          </Vivaldi>
87      </stages>
88  </sandstorm>
```

Appendix B

OIDA Clients

These are the OIDA prototype client scripts, cf. Section 5.4, for data preparation, storage, and lookup.

OIDA Preparation Script

This script `oida_prepare.py` would be run by a publisher to generate databases of encrypted and signed address documents and corresponding keys.

```
1  #!/usr/bin/env python
2  # OIDA-Prepare, Version 0.07
3  # Scenario: We assume a given manufacturer wants to store EPCIS address
       documents
4  # for a range of EPCs for a specific object class.
5  # Note: EPC numbers could also be arbitrary chosen according to a given
       scenario.
6
7  import time
8  import sys
9  import pickle
10 import sha
11 from bsddb import db # Berkeley Database Interface
12 from Crypto.Cipher import AES
13 from Crypto.PublicKey import RSA
14 from Crypto.Hash import SHA
15 from os import urandom
16
17 starttime = time.time() # Start time of experiment.
18
19 # Specific experimental settings:
20 epcdbfilename = './epcdatabase.db' # Name of EPC database file to create.
21 keydbfilename ='./keydatabase.db' # Name of the EPC key database file to create
       .
22 rsafilename = './rsafile' # Name of RSA key file.
23 # EPC structure, here a decimal approximation of example SGTIN-96:
24 emlen = 7 # Dec. EPC manager field length.
25 oclen = 7 # Dec. object-class field length.
```

```
26 selen = 12 # Dec. serial number field length.
27 # EPC range data fields:
28 epcmanager = 5522334 # EPC manager number of EPC range.
29 objectclass = 5667788 # Object-class of EPC range.
30 startserial = 1422003456 # Starting EPC serial number.
31 # Starting EPC as integer:
32 offset = epcmanager * 10**(oclen+selen) + objectclass * 10**selen + startserial
33 k = 2000 # Number of EPCs in range.
34 delimiter = "BEGIN_SIG" # for separating data from signature.
35 # Presentation settings:
36 g = 9 # Rounding time results to g digits after the decimal point.
37
38 # Create RSA key:
39 print "\nGenerating␣RSAkey␣...␣"
40 startgentime = time.time() # Start time of key generation.
41 RSAkey=RSA.generate(2048, urandom)
42 stopgentime = time.time() # Stop time of key generation.
43 print "Done.␣Duration␣of␣Key␣Generation:␣" + str(round(stopgentime -
       startgentime,g)) + "␣seconds."
44
45 rsafile = open(rsafilename, 'w') # In real life, split, distribute the public
       key + certificate from CA.
46 pickle.dump(RSAkey, rsafile)
47 rsafile.close()
48
49 # Create EPC and Key databases.
50 epcdb = db.DB()
51 keydb = db.DB()
52 epcdb.open(epcdbfilename, dbtype=db.DB_BTREE, flags=db.DB_CREATE)
53 keydb.open(keydbfilename, dbtype=db.DB_BTREE, flags=db.DB_CREATE)
54 for epc in range(offset, offset + k):
55         rawdatum = ";;Fictional␣NAPTR␣record␣for␣EPC␣" + str(epc) + """
56 ;;IN _ NAPTR _ order _ pref _ flags _ service _ regexp _ replacement.
57 IN NAPTR 100  50  0 0 u EPC+epcis !^.*$!http://example.com/autoid/cgi-bin/epcis
       .php! . """ + "\nIN␣NAPTR␣100␣␣50␣␣0␣0␣u␣EPC+html␣!^.*$!http://www.example.
       com/prdct/" + str(epc)+ "/info␣."
58
59         #print len(rawdatum) # Use some padding convention to get multiples of
               16 bytes.
60
61         # Key generation from urandom, "The returned data should be
               unpredictable enough for
62         # cryptographic applications, though its exact quality depends on the
               OS ..." (docs.python.org)
63         # AES key must be either 16, 24, or 32 bytes long.
64         epckey = urandom(16)
65         keydb.put(str(epc), str(epckey)) # Insecure local storage of EPC key
               file!
66         obj = AES.new(epckey)
67         cryptdatum = obj.encrypt(rawdatum)
68         hashdatum = SHA.new()
69         hashdatum.update(cryptdatum)
70         hash = hashdatum.digest()
71         signature = str(RSAkey.sign(hash, ""))
72         cryptsigndatum = cryptdatum + delimiter + signature
73         #print len(cryptsigndatum) # Must be lower than 1024 bytes with
               standard Bamboo!
74         epcdb.put(str(epc), cryptsigndatum)
75 epcdb.close()
76 keydb.close()
77
78 endtime = time.time() # End time of experiment.
79 duration = endtime - stopgentime
80 totalduration = endtime - starttime
```

```
81  average = duration/k
82  print str(k) + " documents created, encrypted, and locally stored."
83  print "Total duration: " + str(round(totalduration,g)) + " seconds."
84  print "Storage duration: " + str(round(duration,g)) + " seconds."
85  print "Average: " + str(round(average,g)) + " seconds per document."
86  print "\n"
```

OIDA Publish Script

The following script `oida_put.py` would be run by a publisher to store the encrypted and signed documents to the DHT.

```
1   #!/usr/bin/env python
2   # OIDA-Publish, Version 0.14
3   # Scenario: We assume a given manufacturer wants to store EPCIS address
        documents
4   # for a range of EPCs for a specific object class.
5   # Note: EPC numbers could also be arbitrary chosen according to a given
        scenario.
6
7   import time
8   import sys
9   from bsddb import db # Berkeley Database Interface
10  import sha
11  import socket
12  from xmlrpclib import *
13  import MLab
14
15  # General OIDA settings:
16  gwip = "141.20.103.210" # OIDA gateway to contact.
17  gwport = "15942" # OIDA port for XML-RPC connections.
18  gateway = "http://" + gwip + ":" + gwport +"/"
19  proxy = ServerProxy(gateway) # XML-RPC connection.
20  result = {0:"Success", 1:"Capacity", 2:"Again"} # Status of operation.
21
22  # Specific experimental settings:
23  epcdbfilename = './epcdatabase.db' # Name of EPC database file-
24  keydbfilename ='./keydatabase.db' # Name of the EPC key database file.
25  # EPC structure, here a decimal approximation of example SGTIN-96:
26  emlen = 7 # Dec. EPC manager field length.
27  oclen = 7 # Dec. object-class field length.
28  selen = 12 # Dec. serial number field length.
29  # EPC range data fields:
30  epcmanager = 5522334 # EPC manager number of EPC range.
31  objectclass = 5667788 # Object-class of EPC range.
32  startserial = 1422003456 # Starting EPC serial number.
33  # Starting EPC as integer:
34  offset = epcmanager * 10**(oclen+selen) + objectclass * 10**selen + startserial
35  k = 2000 # Number of EPCs in range.
36  rc = 5 # Number of repliche per EPC (including the first).
37  ttl = 36000 # Global TTL value of the data records to be stored (in sec).
38  timeout = 30 # Connection timeout.
39  socket.setdefaulttimeout(timeout)
40
```

```
41 # Presentation settings:
42 g = 4 # Rounding time results to g digits after the decimal point.
43
44 # Open EPC and Key database files.
45 epcdbread = db.DB()
46 epcdbread.open(epcdbfilename, dbtype=db.DB_BTREE, flags=db.DB_RDONLY)
47 keydbread = db.DB()
48 keydbread.open(keydbfilename, dbtype=db.DB_BTREE, flags=db.DB_RDONLY)
49
50 epcdurationslist = range(k)
51 repdurationslist = []
52 abssuccesscounter = 0
53 failurecounter = 0
54 starttime = time.time() # Start time of experiment.
55 print "\n START"
56
57 # Experiment main loop:
58 for epc in range(offset, offset + k):
59         relsuccesscounter = 0
60         val = Binary(epcdbread.get(str(epc)))
61         epckey = str(keydbread.get(str(epc)))
62         beginepctime = time.time() # Start time of this specific EPC.
63         for r in range(1, rc+1) :
64                 idstring = sha.new(str(epc)+',' + epckey + ',' +str(r)).digest
                        () # sha1("epc,key,r") = Overlay ID
65                 ID = Binary(idstring)
66                 print "Storing document for EPC #" + str(epc - offset + 1) +":
                        " + str(epc) + ", Replica " + str(r)
67                 beginconnecttime = time.time() # Start time of this specific
                        connection.
68                 try :
69                         connectresult = result[proxy.put(ID, val, ttl, "
                                oida_put.py")] # XML-RPC call.
70                 except socket.timeout:
71                                 print "*** Connection Timeout raised and caught
                                        ! Timeout: " + str(timeout) + " ***\n"
72                                 print "*** Attention, no document stored for
                                        EPC " + str(epc)  + ", Replica " + str(r) +
                                        "! ***"
73                                 connectresult = "Failure"
74                 except :
75                                 print "*** Connection Error! ***\n"
76                                 print "*** Attention, no document stored for
                                        EPC " + str(epc)  + ", Replica " + str(r) +
                                        "! ***"
77                                 connectresult = "Failure"
78                 endconnecttime = time.time() # End time of this specific
                        connection.
79                 connectduration = endconnecttime - beginconnecttime # Duration
                        of connection.
80                 print connectresult + ": " + str(round(connectduration,g)) + "
                        seconds."
81                 repdurationslist.append(connectduration)
82                 if connectresult == "Success" : relsuccesscounter += 1
83                 else : failurecounter += 1
84         endepctime = time.time() # End time of this specific EPC.
85         epcduration = endepctime - beginepctime
86         epcdurationslist[epc - offset] = epcduration
87         if relsuccesscounter > 0 : abssuccesscounter += 1
88         if relsuccesscounter < rc : print "*** Attention, only " + str(
               relsuccesscounter) + " repliche for EPC " + str(epc) + " stored! ***
               "
89 endtime = time.time() # End time of experiment.
90 epcdbread.close()
```

```
91  keydbread.close()
92  # Statistics:
93  duration = endtime - starttime
94  average = duration/(k * rc)
95  print "\n"
96  print "OIDA Gateway: " + gwip + ":" + gwport
97  print str(abssuccesscounter) + " out of " + str(k) + " EPCs stored successfully
        (" + str(round((float(abssuccesscounter)/float(k))*100,g+1)) + "%)."
98  print "Statistics for all repliche:"
99  print str(k * rc - failurecounter) + " out of " + str(k * rc) + " repliche
        stored successfully (" + str(round((float(k * rc - failurecounter)/float(k *
         rc))*100,g+1)) + "%)."
100 print "Total duration: " + str(round(duration,g)) + " seconds."
101 print "Median: " + str(round(MLab.median(repdurationslist),g)) + " seconds."
102 print "Average: " + str(round(average,g)) + " seconds."
103 print "Minimum: " + str(round(MLab.min(repdurationslist),g)) + " seconds."
104 print "Maximum: " + str(round(MLab.max(repdurationslist),g)) + " seconds."
105 print "Standard Deviation: " + str(round(MLab.std(repdurationslist),g)) + "
        seconds."
106 print "\n"
```

OIDA Lookup Script

The final script `oida_get.py` would be used by a client to retrieve the encrypted and signed address documents from the DHT, verify the signature by using the publisher public key, and decrypt them.

```
1  #!/usr/bin/env python
2  # OIDA-Retrieve, Version 0.15
3  # Scenario: We assume a given client wants to retrieve EPCIS address documents
4  # for a range of EPCs for a specific object class.
5  # Note: EPC numbers could also be arbitrary chosen according to a given
       scenario.
6
7  import time
8  import sys
9  import pickle
10 from bsddb import db # Berkeley Database Interface
11 import sha
12 from xmlrpclib import *
13 from Crypto.Cipher import AES
14 from Crypto.PublicKey import RSA
15 from Crypto.Hash import SHA # Redundant to sha import above. Could be replaced
       by that.
16 import MLab
17 import socket
18
19 # General OIDA settings:
20 gwip = "141.20.103.211" # OIDA gateway to contact.
21 gwport = "15942" # OIDA port for XML-RPC connections.
22 gateway = "http://" + gwip + ":" + gwport +"/"
23 proxy = ServerProxy(gateway) # XML-RPC connection.
24
25 # Specific experimental settings:
```

```
26 keydbfilename ='./keydatabase.db' # Name of the EPC key database file.
27 rsafilename = './rsafile' # Name of RSA key file.
28 # EPC structure, here a decimal approximation of example SGTIN-96:
29 emlen = 7 # Dec. EPC manager field length.
30 oclen = 7 # Dec. object-class field length.
31 selen = 12 # Dec. serial number field length.
32 # EPC range data fields:
33 epcmanager = 7722334 # EPC manager number of EPC range.
34 objectclass = 5667788 # Object-class of EPC range.
35 startserial = 1422003456 # Starting EPC serial number.
36 # Starting EPC as integer:
37 offset = epcmanager * 10**(oclen+selen) + objectclass * 10**selen + startserial
38 k = 2000 # Number of EPCs in range.
39 rc = 5 # Number of repliche per EPC (including the first).
40 delimiter = "BEGIN_SIG" # for separating data from signature.
41 # Specific Setting: Assume data to be stored already present in memory.
42 # No data diversity neccessary for pure network measurement.
43 # For whole system performance, include local data reading in loop.
44 # Example: From files, from database.
45 maxvals = 10 # Maximum number of documents to return per EPC
46 pm = Binary("") # Pointer to next data document.
47 timeout = 30 # Connection timeout.
48 socket.setdefaulttimeout(timeout)
49 # Presentation settings:
50 g = 4 # Rounding time results to g digits after the decimal point.
51
52 # Import RSA key:
53 rsafile = open(rsafilename, 'r')
54 RSAkey = pickle.load(rsafile)
55 rsafile.close()
56
57 # Open Key database file.
58 keydbread = db.DB()
59 keydbread.open(keydbfilename, dbtype=db.DB_BTREE, flags=db.DB_RDONLY)
60
61 epcdurationslist = range(k)
62 repdurationslist = []
63 documentlist = []
64 abssuccesscounter = 0
65 failurecounter = 0
66 starttime = time.time() # Start time of experiment.
67 print "\n"
68
69 # Experiment main loop:
70 for epc in range(offset, offset + k):
71         relsuccesscounter = 0
72         epckey = str(keydbread.get(str(epc)))
73         beginepctime = time.time() # Start time of this specific EPC.
74         for r in range(1, rc+1) :
75                 idstring = sha.new(str(epc)+',' + epckey + ',' +str(r)).digest
                        () # sha1("epc,r") = Overlay ID
76                 ID = Binary(idstring)
77                 print "Getting␣documents␣for␣EPC␣#" + str(epc - offset + 1) +":
                        ␣" + str(epc) + ",␣Replica␣" + str(r)
78                 beginconnecttime = time.time() # Start time of this specific
                        connection.
79                 while 1:
80                         document = ""
81                         try: vals, pm = proxy.get(ID, maxvals, pm, "oida_get.py
                                ")
82                         except socket.timeout:
83                                 print "***␣Connection␣Timeout␣raised␣and␣caught
                                        !␣***\n"
```

```
84                                      print "*** Attention, no document found for EPC
                                            " + str(epc)  + ", Replica " + str(r) + "! 
                                            ***"
85                                      failurecounter += 1
86                                      document = "NULL"
87                                      break
88                              except :
89                                      print "*** Connection Error! ***" # Includes
                                            any other exception, however.
90                                      print "*** Attention, no document found for EPC
                                            " + str(epc)  + ", Replica " + str(r) + "! 
                                            ***"
91                                      failurecounter += 1
92                                      document = "NULL"
93                                      break
94                              for v in vals:
95                                      cryptic, sigstring = v.data.split(delimiter)
96                                      obj = AES.new(epckey)
97                                      rawdatum = obj.decrypt(cryptic)
98                                      print "Decrypted Data: " + rawdatum + "\n"
99                                      hashdatum = SHA.new()
100                                     hashdatum.update(cryptic)
101                                     hash = hashdatum.digest()
102                                     signature = eval(sigstring) # Trick to convert
                                            from string back to tuple.
103                                     check = RSAkey.verify(hash, signature)
104                                     print "Verifying Signature (1 = OK): " + str(
                                            check)
105                                     if check == 1 :
106                                             document += rawdatum
107                                     else : document += ""
108                             if (pm.data == ""):
109                                     if document == "" :
110                                             failurecounter += 1
111                                             document = "NULL"
112                                             print "*** Attention, no correctly 
                                                    signed document found for EPC " +
                                                    str(epc)  + ", Replica " + str(r) +
                                                    "! ***"
113                                     else : relsuccesscounter += 1
114                                     documentlist.append(document)
115                                     break
116                     endconnecttime = time.time() # End time of this specific
                            connection.
117                     connectduration = endconnecttime - beginconnecttime # Duration
                            of connection.
118                     print "Duration: " + str(round(connectduration,g)) + " seconds
                            .\n"
119                     repdurationslist.append(connectduration)
120
121             endepctime = time.time() # End time of this specific EPC.
122             epcduration = endepctime - beginepctime
123             epcdurationslist[epc - offset] = epcduration
124             if relsuccesscounter > 0 : abssuccesscounter += 1
125             if relsuccesscounter < rc : print "*** Attention, only " + str(
                    relsuccesscounter) + " repliche for EPC " + str(epc) + " retrieved! 
                    ***\n"
126     endtime = time.time() # End time of experiment.
127
128     # Statistics:
129     duration = endtime - starttime
130     average = duration/(k * rc)
131     print "\n"
132     print "OIDA Gateway: " + gwip + ":" + gwport
```

```
133 print str(abssuccesscounter) + " out of " + str(k) + " EPCs retrieved
        successfully (" + str(round((float(abssuccesscounter)/float(k))*100,g+1)) +
        "%)."
134 print "Statistics for all repliche:"
135 print str(k * rc - failurecounter) + " out of " + str(k * rc) + " repliche
        retrieved successfully (" + str(round((float(k * rc - failurecounter)/float(
        k * rc))*100,g+1)) + "%)."
136 print "Total duration: " + str(round(duration,g)) + " seconds."
137 print "Median: " + str(round(MLab.median(repdurationslist),g)) + " seconds."
138 print "Average: " + str(round(average,g)) + " seconds."
139 print "Minimum: " + str(round(MLab.min(repdurationslist),g)) + " seconds."
140 print "Maximum: " + str(round(MLab.max(repdurationslist),g)) + " seconds."
141 print "Standard Deviation: " + str(round(MLab.std(repdurationslist),g)) + "
        seconds."
142 print "\n"
```

Appendix C

Abbreviations

Abbreviation	Extension
ADV	Adversary Set
AES	Advanced Encryption Standard
AS	Autonomous System
API	Application Programming Interface
avg.	Average
B2B	Business-to-Business
B2C	Business-to-Consumer
BGP	Border Gateway Protocol
BIND	Berkeley Internet Name Domain [Daemon]
CA	Certification Authority
CDN	Content Delivery Networks
cf.	confer
CHF	Cryptographic Hash Function
Counter-SH	Set of Counter-Stakeholders
(D)DoS	(Distributed) Denial-of-Service
DHT	Distributed Hash Table
DMZ	Demilitarized Zone
DNS	Domain Name System
DNSKEY	DNS Public Key RR
DNSSEC	DNS Security Extensions
DoS	Denial-of-Service
DS (1)	(EPCIS) Discovery Service
DS (2)	Delegation Signer DNS RR
DSL	Digital Subscriber Line
DTLS	Datagram Transport Layer Security
EAN	European Article Number
ECC	Elliptic Curve Cryptography
e.g.	exempli gratia
EDNS0	Extension Mechanisms for DNS
ENUM	Telephone Number Mapping
EPC	Electronic Product Code
EPCIS	EPC Information Service
et al.	et alii / et aliae
EU	European Union
GB	Gigabyte

GPS	Global Positioning System
GS1	Global Systems One
GTIN	Global Trade Item Number
HIP	Host Identity Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL / TLS
ID	Identifier
IDS	Intrusion Detection System
i.e.	id est
IOT	Internet of Things
IOTNS	IOT Name Service
IP	Internet Protocol
IP	IP Address
IPv4(6)	Internet Protocol Version 4(6)
IPsec	IP Security
ISP	Internet Service Provider
IT	Information Technology
IXP	Internet Exchange Point
KB	Kilobyte
LAN	Local Area Network
LNS	Long Lifetime Neighbor Selection
MB	Megabyte
MAC	Message Authentication Code
mDNS	Multicast DNS
MITM	Man-in-the-middle [attack]
MONS	Multipolar ONS
ms	Millisecond
MX	Mail Exchanger RR
n/a	Not Applicable
NAPTR	Naming Authority Pointer RR
NIS	Network Information Services
NIST	National Institute of Standards and Technology
NSEC	Next Secure RR
OC	Object Class
OID	Object Identifier (generalization of EPC)
OIDA	Object-Information Distribution Architecture
OIS	Object Information Service (generalization of EPCIS)
ONS	Object Naming Service
p.	pagina
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
PDA	Private Database Access
PET	Privacy-enhancing Technologies
PHIDS	Physical Intrusion Detection System
PIR	Private Information Retrieval
PKI	Public-key Infrastructure
PL	PlanetLab
PNS	Proximity Neighbor Selection
POP3	Post Office Protocol Version 3
pp.	paginae
RNS	Random Neighbor Selection
RFC	Request for Comments
RFID	Radio-Frequency Identification

RPC	Remote Procedure Call
RR	Resource Record
RRset	Set of RRs
RRSIG	Resource Record Signature RR
RSA	R. *R*ivest, A. *S*hamir, L. *A*dleman [public-key cryptosystem]
RTT	Round-trip Time
s	Second
SC	Secure Coprocessor
SGTIN	Serialized GTIN
SH	Set of IOT Stakeholders
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
STD	Standard Deviation
s.v.	sub voce
TB	Terabyte
TCP	Transmission Control Protocol
TLD	Top-level Domain
TLS	Transport Layer Security
TTL	Time to Live
TSIG	Transaction Signature
UC	Ubiquitous Computing
UCC	Uniform Code Council
UDDI	Universal Description, Discovery, and Integration
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
US	United States of America
VoIP	Voice over IP
VPN	Virtual Private Network
WINS	Windows Internet Name Service
WLAN	Wireless LAN
w.r.t.	With respect to
WWW	World Wide Web
XML	Extensible Markup Language
XML-RPC	XML Remote Procedure Call

Appendix D

Acknowledgements

I am most grateful to my advisors, my colleagues, fellow researchers, students, friends, and my family for their guidance, cooperation, discussions, and support during my work on this thesis.

In particular, I would like to thank Oliver Günther, Thomas Santen, Seda Gürses, Matthias Fischmann, Sergei Evdokimov, Ignacio Mochales Cuesta, Petra Bulwahn, and Emily Sanford.

And I am also indebted and thankful beyond words to all of my family, especially to Zorro, Nuit, Benji, Nicole, Bernd, and to my mother Luise.

Appendix E

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst und dabei nur die angegebene Literatur sowie die angegebenen Hilfsmittel verwendet zu haben.

Ich bezeuge durch meine Unterschrift, dass meine Angaben über die bei der Abfassung meiner Dissertation benutzten Hilfsmittel, über die mir zuteil gewordene Hilfe sowie über frühere Begutachtungen meiner Dissertation in jeder Hinsicht der Wahrheit entsprechen.

Berlin, 26.06.2008

Benjamin Fabian